DTIC FILE COPY

(2)

AD-A201 207

# NAVAL POSTGRADUATE SCHOOL
# Monterey, California

DTIC
ELECTE
DEC 0 8 1988
S D

D

# THESIS

FORTRAN PROGRAMS FOR AERODYNAMIC
ANALYSES ON THE MICROVAX 2000
CAD CAE WORKSTATION

by

John A. Campbell, Jr.

September 1988

Thesis Advisor J.V. Healey

Approved for public release; distribution is unlimited.

88 12 7 016

Unclassified

security classification of this page

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a Report Security Classification Unclassified | 1b Restrictive Markings |
| 2a Security Classification Authority | 3 Distribution Availability of Report<br>Approved for public release; distribution is unlimited. |
| 2b Declassification Downgrading Schedule | |
| 4 Performing Organization Report Number(s) | 5 Monitoring Organization Report Number(s) |
| 6a Name of Performing Organization<br>Naval Postgraduate School — 6b Office Symbol *(if applicable)* 31 | 7a Name of Monitoring Organization<br>Naval Postgraduate School |
| 6c Address *(city, state, and ZIP code)*<br>Monterey, CA 93943-5000 | 7b Address *(city, state, and ZIP code)*<br>Monterey, CA 93943-5000 |
| 8a Name of Funding Sponsoring Organization — 8b Office Symbol *(if applicable)* | 9 Procurement Instrument Identification Number |
| 8c Address *(city, state, and ZIP code)* | 10 Source of Funding Numbers<br>Program Element No — Project No — Task No — Work Unit Accession No |

11 Title *(include security classification)* FORTRAN PROGRAMS FOR AERODYNAMIC ANALYSES ON THE MICROVAX 2000 CAD CAE WORKSTATION

12 Personal Author(s) John A. Campbell, Jr.

| 13a Type of Report | 13b Time Covered | 14 Date of Report *(year, month, day)* | 15 Page Count |
|---|---|---|---|
| Master's Thesis | From To | September 1988 | 201 |

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 17 Cosati Codes | | | 18 Subject Terms *(continue on reverse if necessary and identify by block number)* |
|---|---|---|---|
| Field | Group | Subgroup | Numerical methods, FORTRAN programs, 2-D, 3-D, Doublet, Panel, Vortex lattice. |
| | | | |
| | | | |


19 Abstract *(continue on reverse if necessary and identify by block number)*

This thesis describes the conversion of four computer programs on the Naval Postgraduate School IBM 3033AP computer system and their implementation on the MicroVax 2000 CAD CAE workstation. The existing 2-D airfoil analysis programs DUBLET and PANEL were extensively modified to improve the user interface. The 3-D wing analysis program VORLAT also received an updated interface. The JETFLAP source program no longer resided on the NPS mainframe and was reconstructed from an original source tape and program listing. This program was then converted from FORTRAN IV for the CDC 6000 series computers to FORTRAN 77 for use on the IBM mainframe and the MicroVAX 2000. An interactive data input program, JETFLAPIN, was developed to simplify data input, provide error checking and correction and thereby enhance the utilization of the JETFLAP program. The programs are intended for use by students in basic and advanced courses in aerodynamics at the Naval Postgraduate School, however they are also applicable to a course in computaional aerodynamics. Keywords: user manuals. (KR)


| 20 Distribution Availability of Abstract | 21 Abstract Security Classification |
|---|---|
| ☒ unclassified unlimited ☐ same as report ☐ DTIC users | Unclassified |

| 22a Name of Responsible Individual | 22b Telephone *(include Area code)* | 22c Office Symbol |
|---|---|---|
| J.V. Healey | (408) 646-2804 | 67He |

DD FORM 1473,84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete — security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

# FORTRAN Programs for Aerodynamic Analyses on the MicroVAX/2000 CAD/CAE Workstation

by

John A. Campbell, Jr.
Lieutenant, United States Coast Guard
B.S., Arizona State University, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1988

Author: John A. Campbell, Jr.

John A. Campbell, Jr.

Approved by: JVHealey

J.V. Healey, Thesis Advisor

E.R. Wood

E.R. Wood, Chairman,
Department of Aeronautics and Astronautics

GESchacher

Gordon E. Schacher,
Dean of Science and Engineering

# ABSTRACT

This thesis describes the conversion of four computer programs on the Naval Postgraduate School IBM 3033AP computer system and their implementation on the MicroVax/2000 CAD/CAE workstation. The existing 2-D airfoil analysis programs DUBLET and PANEL were extensively modified to improve the user interface. The 3-D wing analysis program VORLAT also received an updated interface. The JETFLAP source program no longer resided on the NPS mainframe and was reconstructed from an original source tape and program listing. This program was then converted from FORTRAN IV for the CDC 6000 series computers to FORTRAN 77 for use on the IBM mainframe and the MicroVAX/2000. An interactive data input program, JETFLAPIN, was developed to simplify data input, provide error checking and correction and thereby enhance the utilization of the JETFLAP program. The programs are intended for use by students in basic and advanced courses in aerodynamics at the Naval Postgraduate School, however they are also applicable to a course in computaional aerodynamics.

## THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

## TABLE OF CONTENTS

## LIST OF FIGURES

## ACKNOWLEDGEMENTS

First I would like to thank Professor Val Healey, who provided me with the idea for this project and greatly assisted me in understanding the theory underlying these programs. I wish also to express my thanks to the staff of the W. R. Church Computer Center, in particular to Ms. June Favorite, for her assistance in the transfer of the JETFLAP programs from magnetic tape, to Mrs. Jane Kretzmann, for her patience and valuable insights on the use of DISSPLA, and to Mr. Larry Frazier, whose efforts with the Script processor made this publication possible. An additional note of thanks go to Ms. Rosalie Johnson, of the Computer Science Department, and Mr. David Marco of the Mechanical Engineering Department, for sharing their technical expertise and transferring the numerous files contained in this work between the VAX and the IBM mainframe computers. Not to be forgotten is the special contribution made by Professor E. M. Wu, by providing me with the use of his lab and MicroVax workstation for the program conversion. Most of all, I want to thank my children, Nicholas, John, Colin and Elaina, for their love and understanding during this effort and my dear wife, Juleen, whose personal strength and continual support enabled me to bring this project to fruition.

# II. INTRODUCTION

Current aerodynamic analysis relies heavily upon numerical methods for estimating the aerodynamic coefficients of airfoils and wings. This thesis was undertaken to provide the students of the aeronautical engineering curriculum with a series of computer programs that would give them a better appreciation and understanding of several computational methods that have been applied to classical aerodynamic theory.

The Department of Aeronautics and Astronautics at the Naval Postgraduate School (NPS), in conjunction with the Mechanical Engineering Department, is developing a computer-aided design computer-aided engineering (CAD/CAE) laboratory for use in research and teaching by their respective curriculums. The system is based on a network of Digital Equipment Corporation (DEC) MicroVAX 2000 workstations. There is an ongoing requirement to provide specialized software (programs) for the computer network that is usable by the students and staff to support current and future courses and research.

At the time of this writing, several aerodynamic analysis programs reside on the NPS IBM 3033AP mainframe computer. They are in various states of repair[1] and due to constant software and hardware upgrades of the mainframe system some programs provide limited output capabilities[2] while others are becoming unusable due to compiler changes. There is also a wide range in the amount and quality of the documentation available for each program. This thesis seeks to remedy a portion of this problem and support the previously mentioned software needs requirement by providing a set of baseline programs and thorough documentation which will extend the life of these valuable programs and allow further upgrades and eventually the incorporation of graphics routines by future users.

The programs contained in this work were selected on the basis of their applicability to the present courses taught in basic and advanced aerodynamics at NPS, the documentation available and previous user inputs. They were revised or modified with the

---

1 Source code is not available for some programs, in particular FLO27. Since certain output flags for FLO27 were set in the source code and the user was unable to alter these, an inordinate amount of unwanted output was produced.

2 Several programs, notably FLO27, JETFLP and those used in the Aircraft Combat Survivability and Lethality courses have lost their graphical output due to software incompatibility problems.

intent that they be used for preliminary design and to evaluate the changes in aerodynamic coefficients due to changes in one or more of the input parameters. To this end, the following factors were empasized in modifying or creating the programs to make them easily understood and utilized:

- Error checking/correction capability.
- Capability to make multiple runs in one session.
- Capability to change one or more parameters on subsequent runs.
- Utilize a standardized interface (to the extent possible).
- Allow user defined names for input/output files.

This document briefly describes the basic theory behind the 2-D airfoil and 3-D wing analysis programs and the reprogramming required for transfer and conversion of the selected programs from the IBM 3033AP and CDC 6000 series computers to the MicroVAX/2000 CAD/CAE workstation.

A users manual for each program is contained in the appendices. These provide a short discussion on the purpose of the program, input requirements and constraints, program operation and the program output. A sample input session, input data file (if required) and the resulting output as well as a complete program listing is also included.

Project results and recommendations for future work are given.

# III. BASIC THEORY OF 2-D AIRFOIL ANALYSIS PROGRAMS

## A. INTRODUCTION

The following sections are intended to present the reader with a basic understanding of the ideal fluid flow concepts underlying the 2-D airfoil analysis programs. This brief summary contains just a few highlights which would be obtained from a course in the fundamentals of aerodynamics and in no way attempts to provide the reader with a firm foundation in aerodynamics or fluid flows.

It will be assumed that the reader is familiar with the concepts of velocity potential, $\phi$, stream function, $\psi$, and their derivatives. It is further assumed that the reader has some familiarity with the concepts of the basic fluid flows: uniform stream, source, sink, vortex and doublet. Figure 1 depicts these basic fluid flows and provides and example of how two of these flows, a uniform stream and a doublet, may be combined to model the flow over a cylinder. A thorough discussion of these flows and their properties may be found in most aerodynamics texts. References 1, 2, 3 and 4 were instrumental in the preparation of the following sections.

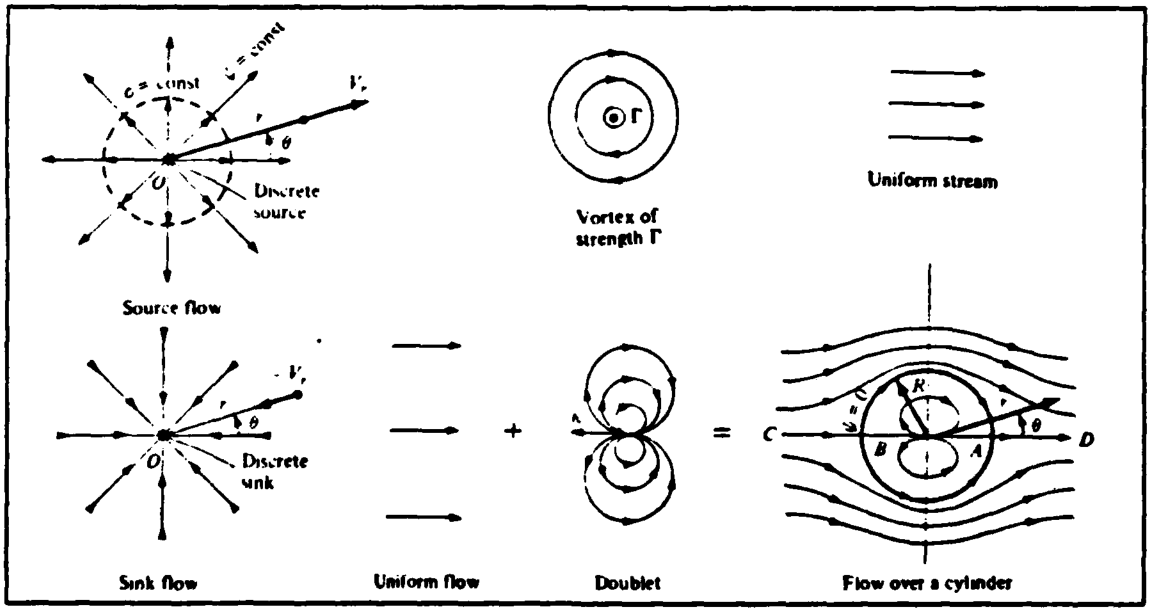


**Figure 1. Basic Fluid Flows**

## B. PROGRAM DUBLET

The type of analysis used here is a *direct* method in which the shape of an ellipsoid or airfoil-like body is specified and the problem is to solve for the distribution of singularities which, in combination with a uniform stream, produce the flow over the body.

This program provides a numerical method for approximating the solution of the integral equation for the line doublet distribution for a symmetrical airfoil at zero lift in incompressible irrotational flow. With the doublet strength known, the velocity field can be determined using equations for the stream function and velocity potential. Once the velocity field is known, the pressure field may be determined using the Bernoulli equation.

For this problem the airfoil body shape is specified as $y = Y(x)$. It is a closed form which has a finite length or chord, $c$ as shown in Figure 2.

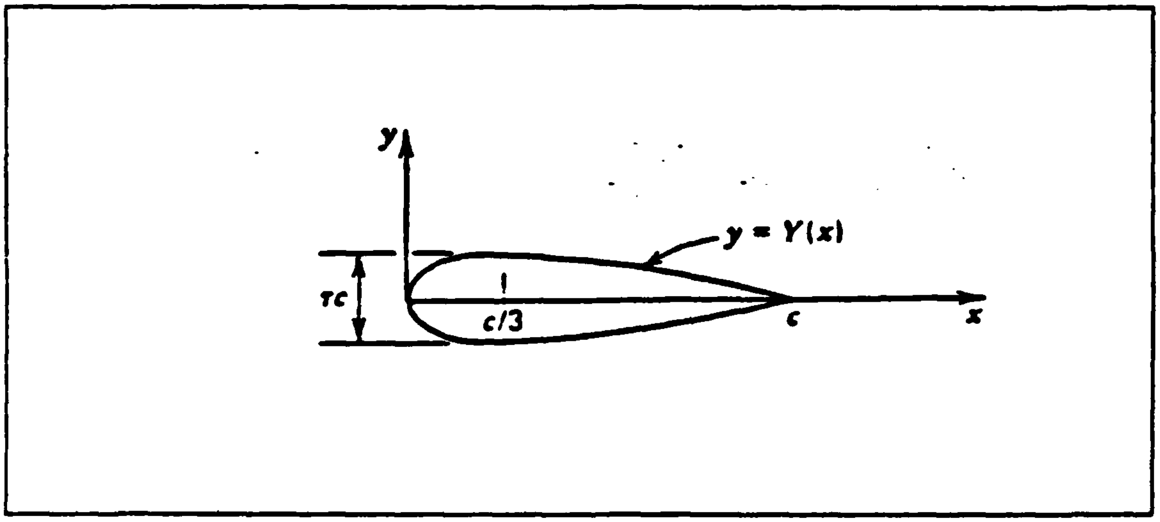


**Figure 2. An Airfoil-like Shape**

Such a shape can be defined by an equation of the form:

$$Y(x) = A\sqrt{\frac{x}{c}}\,(c - x) \tag{1}$$

This shape is to be modeled by a string of doublets along the $x$ axis, and the strength of each doublet to be determined. The solution is required to meet the flow tangency condition and the doublets are required to be within the body.

Since thin-airfoil theory fails near the stagnation points and it is not physically possible for the source distribution to extend to the ends of the body and still meet the

flow tangency condition, there must exist a finite distance between the ends of the source distribution and the stagnation points.

This distance is determined by approximating the shape of a blunt-nosed airfoil near its nose, $x = 0$, as parabolic. Using thin-airfoil theory and the radius of curvature (Figure 3), the source strength near the leading edge of the source distribution can be approximated. Applying this approximation and the requirement that the source-induced velocity must cancel that of the onset flow at the stagnation point, it can be shown [Ref. 1 pp.52-54], that the separation distance between the stagnation point and the leading edge of the source distribution is approximately half the radius of curvature of the nose of the body. A similar analysis holds for the other end of the body.

$$\varepsilon \approx \frac{\rho}{2}$$

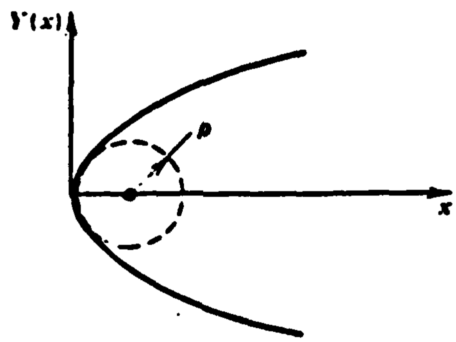


Figure 3. The Radius of Curvature of a Leading Edge

The program DUBLET incorporates the half radius of curvature inset and satisfies the flow tangency requirements using an iterative approach. This is done by taking an approximation to the proper inset, solving the set of simultaneous equations for the doublet strength distribution which satisfies the flow tangency condition and then evaluating the resulting velocity at the stagnation points. If the velocity is not sufficiently close to zero, the estimated values are revised and the process is repeated. The iterative approach used is an interval-halving or bisection method of root-finding similar to that described in Ref. 5.

A more complete development of the thin-airfoil theory and the underlying equations used to derive this method are detailed by Moran [Ref. 1].

## C. PROGRAM PANEL

This analysis again uses the *direct* method to solve for the proper distribution of singularities[3] on a body which, in combination with a uniform stream, provide the flow over the body.

This program uses a numerical approach to provide an approximation to the solution of the integral equation for the source and vortex distribution on the surface of a lifting body in incompressible irrotational flow. It is specifically designed to evaluate NACA four- digit airfoils and NACA five-digit airfoils of the 230XX series; however provisions are made within the program for entry of any arbitrary airfoil shape.

The following presents some reasons for and a brief development of the panel method. Although thin airfoil theory gives reasonably good results for lift and moment coefficients, it ignores the effect on those coefficients of the thickness distribution. In addition, thin airfoil theory gives good pressure distribution results only away from the stagnation points. Since proper design of an airfoil requires an accurate prediction of its pressure distribution, more powerful methods are based on the distributions of sources and vortices or doublets. This is empasized by Moran when he states

> "To avoid the inaccuracies of thin-airfoil theory, the flow-tangency condition must be satisfied on the body surface and...the singularities should be distributed on the body surface rather than on the chord line or any other line within or without the body."

To achieve this placement of the singularities on the body, the body surface is approximated by a collection of straight line *panels*. This form of surface approximation is where the panel method receives its name. Program PANEL uses a solution method based on sources and vortices distributed on these surface panels.

The potential for this flow may be described as

$$\phi = \phi_\infty + \phi_S + \phi_V \tag{2}$$

where $\phi_\infty$ is the potential of the uniform onset flow, which can be written in a Cartesian system as

$$\phi_\infty = V_\infty x \cos\alpha + V_\infty y \sin\alpha \tag{3}$$

---

3 "Singularities" is used here as a generic term for sources, vortices, doublets and other fundamental solutions of the Laplace equation that blow up--are "singular"--at some point outside the flow field.

where $V_\infty$ is the velocity of the uniform flow, and $\alpha$ is the angle between the flow direction and the $x$ axis. The remaining potential terms are defined as

$$\phi_S \equiv \int \frac{q(s)}{2\pi} \ln r \, ds \tag{4}$$

$$\phi_V \equiv -\int \frac{\gamma(s)}{2\pi} \theta \, ds \tag{5}$$

in which the integrations are over the body surface. This defines $\phi_S$, as the potential of a source distribution of strength $q(s)$ per unit length and $\phi_V$, as the potential of a vortex distribution of strength $\gamma(s)$ per unit length. Figure 4 shows that $s$ is the distance measured along the surface from some arbitrary reference point--in this case the leading edge has been chosen--to the "field point", $(x,y)$, or $(r, \theta)$ in polar coordinates.

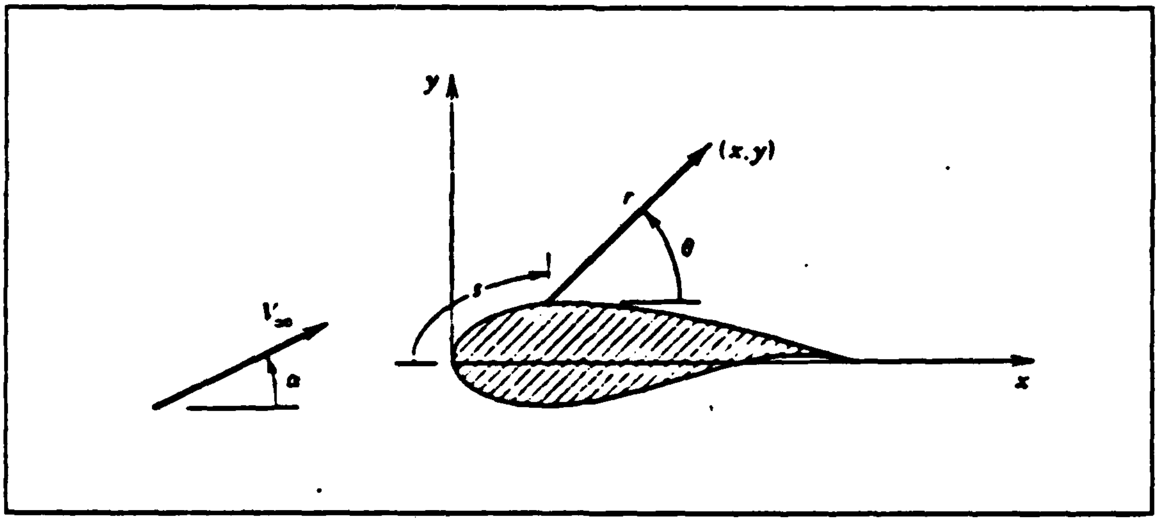


**Figure 4. Nomenclature for the Analysis by the Panel Method**

We seek a solution where $q(s)$ and $\gamma(s)$ are determined so as to meet the boundary condition of flow tangency and the Kutta condition. The latter is the requirement that the stagnation point be at the trailing edge[4].

The view of this problem taken by Hess and Smith [Ref. 6] is that the source strength governs the flow tangency condition and the Kutta condition governs the

[4] All airfoils considered here are assumed to have sharp trailing edges.

vortex strength[5]. They make the simplifying assumption that the vortex strength is taken constant over the whole airfoil, i.e. $\gamma(s) = \gamma$ , and justify this by reasoning that, since the Kutta condition governs the vortex strength, and the Kutta condition involves only the trailing edge, then the vortex strength can be represented by a single number. Conversely, the source strength must vary over the surface to allow the flow tangency condition to be satisfied at all points on the body surface.

The integrals of equations (4) and (5) are difficult to evaluate unless the surface on which the singularities are distributed is a straight line. This is where the surface panels come into play. The body is divided up into a set of panels by selecting a set of $N$ points, called *nodes*, which are then connected by straight lines. This results in an approximation of the body composed of $N$ nodes and panels as shown in Figure 5.

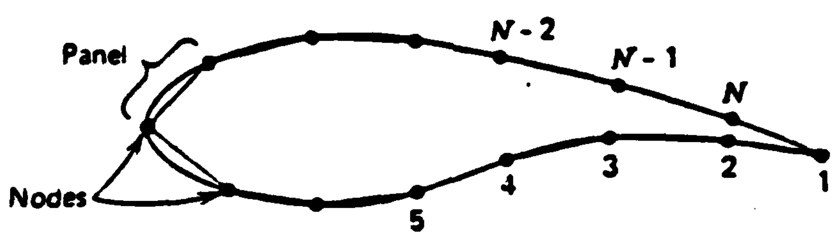


**Figure 5. Definition of Nodes and Panels**

The sources and vortices are distributed on the straight line panels and the constant vortex strength assumption is incorporated so that the potential given by equation (2), as developed in equations (3) through (5), may be written as:

$$\phi = V_\infty x \cos\alpha + V_\infty y \sin\alpha + \sum_{j=1}^{N} \int_{\substack{panel \\ j}} \left[ \frac{q(s)}{2\pi} \ln r - \frac{\gamma}{2\pi} \theta \right] ds \qquad (6)$$

[5] In actuality, both singularity distributions are important in satisfying either condition.

To allow evaluation of the integrals in equation (6), the source strength is taken to be constant on each panel, but allowed to vary from panel to panel, i.e.

$$q(s) = q_i \text{ on panel } i, \quad i = 1,\ldots,N \tag{7}$$

The parameters to be determined are then the $N$ source strengths $q_i$ and the single vortex strength $\gamma$. These are found by imposing the flow tangency condition at $N$ control points and a corollary to the Kutta condition which states, "Near the trailing edge, the flow speeds on the upper and lower surfaces of the airfoil are equal at equal distances from the trailing edge."[Ref. 1]

Moran [Ref. 1] provides a clear explanation of the geometric development of the problem and the resulting set of $N + 1$ equations in the unknowns $q_i$, $i = 1,\ldots,N$, and $\gamma$. This leads into a discussion regarding the development of a FORTRAN program that uses the panel method. Program PANEL sets up and solves this set of equations. The tangential velocity at the midpoint of each panel is then evaluated and its associated pressure coefficient $C_p$ is calculated. By assuming the latter to be constant over each panel, the estimated lift and moment may then be calculated.

# IV. BASIC THEORY OF 3-D WING ANALYSIS PROGRAMS

## A. INTRODUCTION

As discussed in the previous section on 2-D airfoil theory, there are several ways to model the source of forces acting on a body surrounded by a moving fluid. These included potential functions, vortex distributions, circulation distributions and pressure differential distributions. These models are related to one another and each has advantages and disadvantages. Both of the following programs, VORLAT and JETFLAP, rely on a distribution of discrete horseshoe vortices to model the flow over a wing.

### 1. The Horseshoe Vortex

To provide the reader with an understanding of the theory behind the VORLAT and JETFLAP programs, it is necessary to explain what a horseshoe vortex is and what properties it has. References 1, 2, and 3 provided a basis for much of the material contained in this section.

The idea of th horseshoe vortex was developed by Prandtl and Lanchester while trying to provide a simplified model of the ideal flow over a wing. Prandtl reasoned that a vortex filament of strength $\Gamma$, bound to a fixed location in a flow--a bound vortex--will experience a force $L = \rho_\infty V_\infty \Gamma$ from the Kutta-Zhukovsky theorem. To satisfy Helmholz' theorem which states that a vortex filament cannot end in a fluid, the vortex filament continues as two free vortices extending downstream from the wing tips to infinity. The construction of this vortex is in the shape of a horseshoe and it is therefore called a *horseshoe vortex*. It is correctly pointed out however by Zucker that, "...the word "horseshoe", although in common usage, is misleading since these filaments are actually "closed" back at the place where the motion originated."[Ref. 3]

As shown in Figure 6, the wing is replaced by a "lifting line" perpendicular to the flight direction and located at the quarter-chord, with the two free vortices trailing from the wing tips.

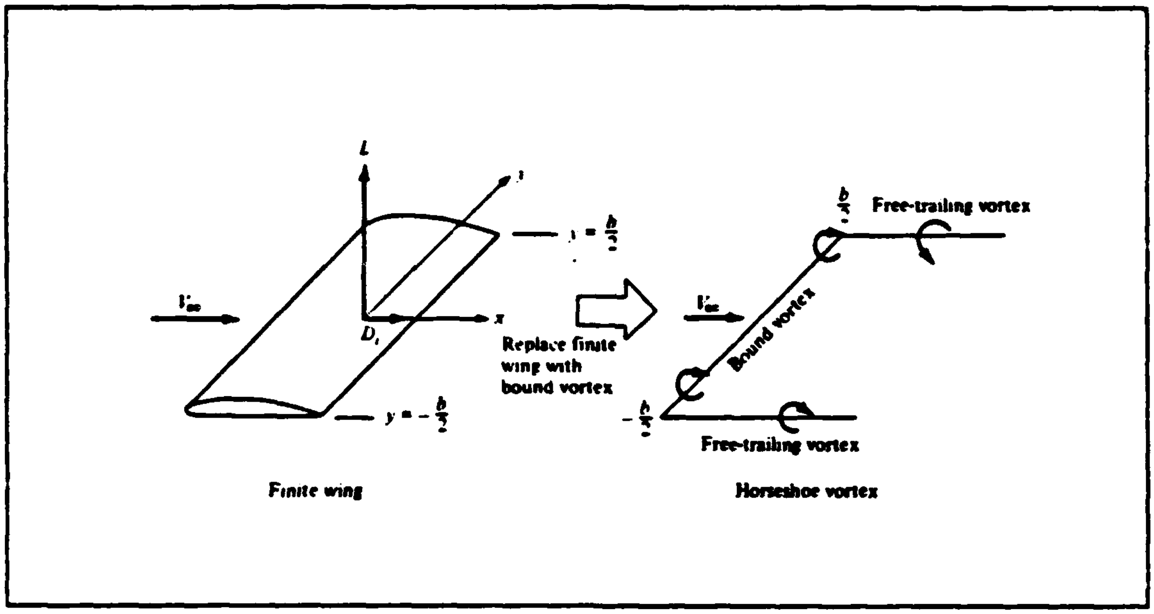


**Figure 6. Replacement of the Finite Wing with a Bound Vortex [Ref. 2]**

This model did not provide a very realistic simulation of the downwash distribution of a finite wing; especially near the tips where the predicted downwash approaches an infinite value. The downwash distribution as a function of the span, $w(y)$, is shown in Figure 7.

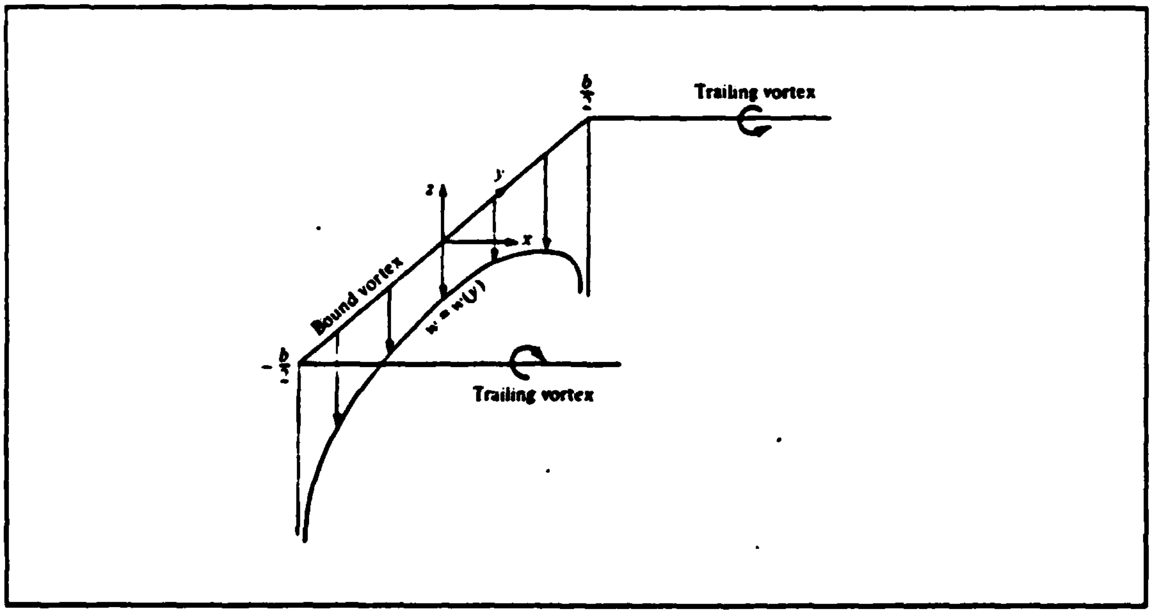


**Figure 7. Downwash Distribution Along $y$ Axis for a Horseshoe Vortex [Ref. 2]**

An improvement on this model was the "lifting line" model which superimposed a large number of horseshoe vortices. each with a different length of bound vortex, but with all the bound vortices lying along a single line. This is depicted in Figure 8 which has three horseshoe vortices of strengths, $\Delta\Gamma_1$, $\Delta\Gamma_2$ and $\Delta\Gamma_3$. The variation of $\Gamma$ along the lifting line is denoted by the vertical bars. Since $L \propto \Gamma$, this is also an indication of the lift distribution. It should be noted that the strength of each trailing vortex is equal to the change in circulation along the lifting line at the point where the trailing vortex starts.

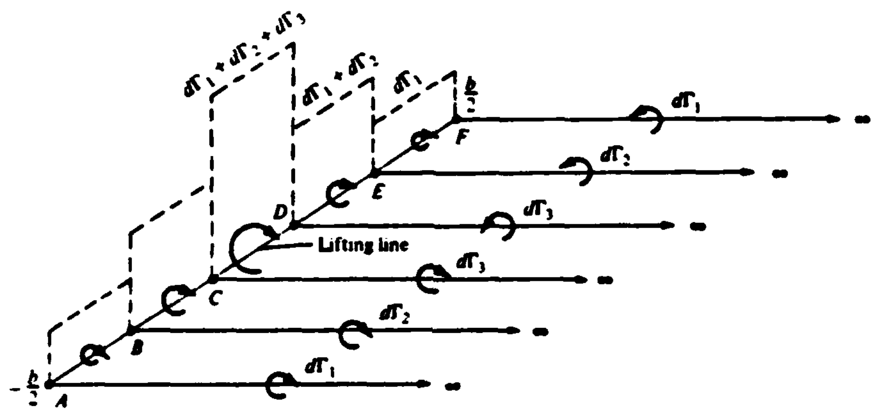


Figure 8. Superposition of Three Horseshoe Vortices Along a Lifting Line [Ref. 2]

This model is good for high aspect ratio straight wings and provides an excellent prediction of spanwise loading and overall lift. It cannot however, produce chordwise pressure distributions and moment data.

To deal with low aspect ratio straight wings, the model is extended by placing a *series* of lifting lines on the plane of the wing at different chordwise stations, all parallel to the $y$ axis. In the limit of an infinite number of these lifting lines, we obtain a vortex sheet, where the vortex lines run parallel to the $y$ axis. The strength of the sheet per unit area is denoted by $\gamma$, where the latter varies in the $y$ direction in a manner analogous to the variation of $\Gamma$ for the single lifting line. In addition, each lifting line will have, in

general, a different overall strength, so that $\gamma$ also varies with $x$. This relation, $\gamma = \gamma(x,y)$ is shown in Figure 9.

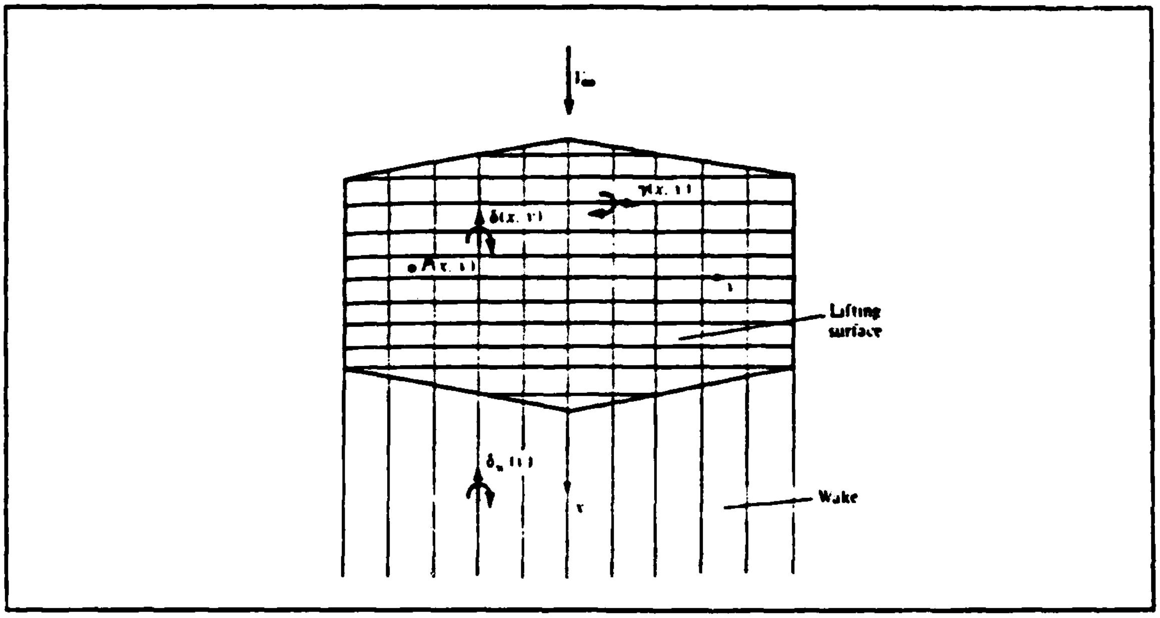


**Figure 9. Schematic of a Lifting Surface [Ref. 2]**

This vortex sheet results in a *lifting surface* distributed over the entire planform of the wing. The strength of the lifting surface at any point on the surface is given by $\gamma = \gamma(x,y)$. The aim of the lifting surface theory is to find $\gamma(x,y)$ such that the flow-tangency condition is satisfied at all points on the wing.

For computational purposes the planform is divided into a finite number of square or rectangular panels and the *ij*th panel chosen for initial computation. The spanwise vorticity on each panel is assumed to be concentrated at the quarter-chord point of the panel and the flow tangency condition is satisfied at the "control point" which is located at the three-quarter chord point of the panel [Ref. 1]. The wing problem then reduces to computation of the velocity at the control point on this *ij*th panel due to all the other panels. This velocity is combined with the freestream value and the tangency condition applied. For each panel, there is therefore, one linear equation and with $N$ panels there are $N$ such equations. Matrix methods are applied to solve this system and with the vorticity distribution known, the Kutta-Zhukovsky theorem is applied to obtain the lift and moments. The induced drag can be computed from the downwash, which is known at the control points. The vortex-lattice method used by program VORLAT is a simple approach used to solve for $\gamma = \gamma(x,y)$.

## B. PROGRAM VORLAT

Program VORLAT implements the vortex-lattice method to determine the solution for the vortex strength distribution on a flat, untwisted, rectangular wing. A set of horseshoe vortices are used to approximate the flow over a wing of low aspect ratio. This is a version of the VORLAT program by Moran [Ref. 1] which has been highly modified and now incorporates a cosine spacing scheme.

The User's Manual presents a short description of the VORLAT program. For complete coverage of the original VORLAT program, consult Moran [Ref. 1].

## C. PROGRAMS JETFLAP AND JETFLAPIN

Program JETFLAP was written by M. L. Lopez, C. C. Shen, and N. F. Wasson at the Douglas Aircraft Company, Long Beach, California. The program is based on A Theoretical Method for Calculating the Aerodynamic Characteristics of Jet-Flapped Wings [Ref. 7] which was developed under a research contract sponsored by the Office of Naval Research. The program is quite extensive and has the capability of determining the following aerodynamic characteristics of wings of arbitrary planform:

- Spanwise and chordwise loading
- Spanwise variation of induced drag
- A capability to investigate the effects of:
  - Part span flaps
  - Part span blowing
  - Pitching, rolling, yawing and sideslip
- Total lift and induced drag (momentum method), pitching, yawing and rolling moments, etc.

The program also provides the capability to investigate the effects of a variation of leading and trailing flap deflection, camber, twist, jet deflection and jet momentum.

Despite the many capabilities of this program and the revised User's Manual developed by Soderman [Ref. 8] in 1976, the program has had limited use at the Naval Postgraduate School since then. This author feels that a major reason for its lack of use is the inordinate amount of time required for the user to prepare and input the data file for even the most elementary planform.

To alleviate this problem, the author developed Program JETFLAPIN, an interactive data entry program to interface with the JETFLAP program. To ensure compatibility, much of JETFLAPIN was created using existing subroutines from JETFLAP.

The JETFLAPIN program provides the user with a method of developing an almost error-free[6] input data file for use with the main JETFLAP program.

The JETFLAPIN program provides error-checking, data review, correction, assurance that all required data has been entered and the elimination of redundant data entry.

[6] While it is still possible for the user to input bad data values, the errors due to values out of limits or incorrect formatting have been virtually eliminated.

# V. PROGRAM TRANSFER AND CONVERSION

## A. INTRODUCTION

This section discussed the steps taken in the transfer of the programs DUBLET, PANEL, VORLAT and JETFLAP from the IBM mainframe computer and their conversion for use on the MicroVAX/2000 workstation. The information provided here will be of use to others planning future transfers of programs between the IBM mainframe computer and the MicroVAX/2000.

## B. FILE TRANSFER

### 1. Programs DUBLET, PANEL, VORLAT

The programs DUBLET, PANEL and VORLAT, were located on the IBM mainframe under the user account 4632P, which was set up for use by the Numerical Methods course, AE 4632, taught in the Aeronautical Engineering Department.[7] Each program was opertational on the mainframe and was activated through the use of an executive calling routine referred to as an "EXEC". These EXECs and the program source code files were readily available for transfer.

Each program and its calling EXEC were transferred to the VAX 11/780 located in the Computer Science Department. This was necessary as the AE/ME VAX network is not currently linked with the IBM mainframe. This transfer was conducted by Mr. David Marco, a computer technician working on the AE/ME VAX system, using the VAX 2780/3780 Protocol Emulator. The file transfer procedures outlined in a Computer Science Department handout covering the RJE File Transfer Package were followed. When the transfer was completed, the files were downloaded to a magnetic tape cartridge, a DEC TK50.

The tape was then taken to the MicroVAX/2000 workstation and loaded into the DEC TK50 tape drive subsystem connected to the workstation. The files were then tranferred from the tape to the workstation's hard disk. From here the files could be edited using the VAX EDT editor [Ref. 9], compiled, linked and run under VAX FORTRAN version V4.0.

---

[7] The read-only password for this account is JVH.

### 2. Program JETFLAP

Program JETFLAP had to be handled quite differently than the other programs. It too was operational on the mainframe, however it had been converted into a cataloged procedure, JETFLP, and was executed using a Job Control Language (JCL) routine. An example of this JCL file is shown in Figure 10. More information on how to create and use JCL files may be found in the User's Guide to MVS at NPS [Ref. 10] or the IBM JCL User's Guide [Ref. 11].

```
//TAPER JOB (1461,1478),'DOUGLAS JETFLAP PRGM',CLASS=C
//*MAIN ORG=NAVPGS.1461P
//  EXEC JETFLP
//SYSIN DD *
Tapered Swept Wing, AR=8, Sweep Angle 45, 10X10 W/Semi-Circle Spacing
50.0000   20.000    0.0         10.43       10.43
1001000001020000
.993844   .969372   .921032   .850012   .758062   .647446   .520888
.381504   .232726   .078217
01010101010101010101
10
.000000   .024472   .095492   .206107   .345492   .5000     .654508
.793893   .904508   .975528
8.0       45.0      0.45
9
/*
//
```

**Figure 10. Sample JETFLAP Batch JCL File**

After an exhaustive search by the personnel of the W. R. Church Computer Center at the NPS, it was determined that only the compiled version of the program existed on the IBM mainframe. The source code had been purged from the system and was not recoverable.

A magnetic tape containing the original Douglas Aircraft Company program was obtained from Dr. M. F. Platzer. This copy had been obtained during thesis work conducted by LCDR A. P. SODERMAN. The tape was logged into the NPS computer center and its characteristics were determined using the tape scan JCL utility shown in Figure 11.

```
//JETFLP   JOB (1461,9999),'JETFLP TSCAN1',CLASS=E
//*MAIN SYSTEM=SY2,RINGCHK=NO
//*
//*   Print tape file  characteristics for any tape
//*
// EXEC TSCAN,VOLIN=JETFLP,DCBIN='DEN=2',UNITIN='3400-4'
//
```

**Figure 11. Magnetic Tape Scan Utility (TSCAN) JCL File**

The tape scan utility revealed that the tape used a very old tape density of 800 BPI. The computer center still had an 800 BPI magnetic tape machine, however they recommended that the contents of this tape be copied onto a new tape using the more common density of 1600 BPI. This was accomplished using the magnetic tape copy utility JCL file shown in Figure 12. The name of the original tape volume was JETFLP. This was changed to JTFLAP on the new copy.

```
//JCOPY   JOB (1461,9999),'JETFLP COPY',CLASS=E
//*
//*   COPY TAPE FILE FROM 800BPI TO ANOTHER AT 1600BPI
//*
// EXEC TAPE,VOLIN=JETFLP,DCBIN='DEN=2',UNITIN='3400-4',
//       VOLOUT=JTFLAP
//SYSIN  DD  *
CPYEND(10,11)
/*
//
```

**Figure 12. Sample Tape Copy JCL File**

Several parity errors occurred while reading the original tape during the copying process. This was an indication that the files contained on the original tape or those obtained through the transfer process may contain errors.

To discover the contents of the tape, a magnetic tape dump utility JCL file was used. This file is shown in Figure 13.

```
//JTFLAP    JOB (1461,9999),'JTFLAP TAPE1',CLASS=E
//*MAIN SYSTEM=SY2,RINGCHK=NO,LINES=(10)
//*
//*   PRINT THE CONTENTS OF EVERY FILE ON THE TAPE.
//*
// EXEC TAPE,VOLIN=JTFLAP
//SYSIN DD *
DMPFIL(10,256,1)
/*
//
```

**Figure 13. Sample Tape Dump Utility JCL File**

A quick review of the printout of produced by this utility revealed that the tape did contain a complete set of the desired files and these were transferred to the author's working disk (A disk) using the procedures outlined in Reference 12. The JCL file used to perform the transfer from tape to the mainframe is shown in Figure 14.

```
//JTFLAP JOB (1461,9999),'JETFLAP TRANSFER',CLASS=A
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,DSN=MSS.C0052.JETFLP
//SYSUT2 DD UNIT=3350,VOL=SER=MVS004,DISP=(NEW,KEEP),
//          SPACE=(CYL,(1,1)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),
//          DSN=S1461.JETFLP
//
```

**Figure 14. Sample Tape Transfer JCL File**

The file was edited to remove the extra lines associated with the transfer process, specifically header, trailer and system information lines associated with the JCL tape transfer utility. The transfer process also places the record number and record length at the beginning of each record. These were removed. The edited version of the program consisted of 4661 lines of FORTRAN code.

Examination of the program file revealed that erroneous pieces of information[8] had crept into the source file. These were due to either the transfer process itself or effects of the environment and aging on the magnetic tape. Regardless of their source, these errors corrupted the source code to such a degree that it would not compile properly on the mainframe.

An attempt to compile the program was made using the VS FORTRAN compiler with its extended error messages[9] to locate as many errors as possible. The listing which was produced flagged all critical areas of the program which required revision. Corrections to the program were made using the listing as a guide. Corrections to non-critical areas of the program, such as comment lines, were made using the program source code listings contained in References 8 and 7 as guides.

It was noted during the editing process that no further errors were encountered in the program following line 2462. This leads the author to the conclusion that the errors were not due to the transfer process, but solely due to defects present in the outer windings of the source tape.

Following completion of the editing process, the program was compiled satisfactorily. Since the program was written using several commands specific to FORTRAN IV it was necessary to compile the program using the (LVL(66)) option with the VS FORTRAN compiler. This invokes the FORTRAN IV version of the VS FORTRAN compiler which allows proper interpretation and compilation of older programs written under the FORTRAN IV standard.

The successfully compiled JETFLAP program was then run using the sample data files provided in References 8 and 7 as input files. The results were then compared to those tabulated in References 8 and 7 which were obtained using the same data files. A slight difference was discovered between the computed values for the moment coefficients (CM and CMG). This difference was traced to a program line for CMG(K) in Subroutine SLOAD which had been modified in [Ref. 8: p. 338] and [Ref. 7: p. C-19], but had not been corrected on the version of the program contained on the source tape. Modification of this line and subsequent compilation and running of the program produced results identical to those contained in References 8 and 7. An additional

8 The erroneous data consisted of extra spaces, non-standard characters and improperly interpreted characters, i.e., several O's were interpreted as M's.

9 The WATFIV compiler is more thorough and produces an even greater number of messages. It is recommended for use on smaller programs or in the final stages of program development due to its extensive output.

comparison was made with the data file and results produced by S. M. White, as part of a class project for AE 3501[Ref. 13]. Again the results were identical. It was then felt that the program was ready to be ported over to the MicroVAX/2000.

The JETFLAP program was transferred from the IBM mainframe to the MicroVAX/2000 in the same manner described previously. It was compiled using the /NOF77 qualifier under VAX FORTRAN and appeared to compile successfully. When a sample run was executed, the program terminated abnormally. This began an extended period of debugging to achieve proper operation of the program on the MicroVAX/2000.

## C. CONVERSION AND REPROGRAMMING

### 1. Programs DUBLET, PANEL, VORLAT

The programs DUBLET, PANEL and VORLAT were written to FORTRAN 77 standards and therefore required little modification to become operational on the MicroVAX/2000. The only significant changes required involved the handling and assignment of input and output data files. As discussed in the section on file transfer, each of these programs had an EXEC file which related to it. Each EXEC contained the name of the program to be run and its associated *file definition* statements. The file definition statements, FILEDEFs, assign input/output devices and were used to define input and output file names and attributes and associate these with the logical unit numbers[10] assigned in the called program. An example of these FILEDEFs, with the FILEDEF command abbreviated to FI, are shown in Figure 15. More information on these may be found in the User's Guide to VM/CMS at NPS [Ref. 14] or the IBM CMS Command Reference [Ref. 15].

10 A logical unit number is specified or implied as part of the I/O statement and it designates the device or file to or from which data is transferred. Logical unit numbers are integers from 0 to 99.

```
&TRACE ON
FI 1 DISK JTFLAP DAT1 B (RECFM F LRECL 2400 BLKSIZE 2400 DSORG DA
FI 2 DISK JTFLAP DATA2 B (RECFM VBS LRECL 860 BLKSIZE 3460
FI 3 DISK JTFLAP DATA3 B (RECFM VBS LRECL 860 BLKSIZE 3460
FI 4 DISK JTFLAP DATA4 B (RECFM VBS LRECL 860 BLKSIZE 3460
FI 5 DISK JTFLAP DATAIN (PERM
FI 6 DISK JTFLAP DATAOUT (RECFM FBA LRECL 133 BLKSIZE 3325
GLOBAL TXTLIB VFORTLIB CMSLIB
LOAD JTFLAP
GLOBAL LOADLIB VFLODLIB
CLRSCRN
START *
&TYPE COMPUTING PROCESSING IS COMPLETED
```

**Figure 15. FILEDEFs in a Sample JTFLAP EXEC File**

Although the use of EXEC files and FILEDEFs is relatively easy and is common practice on the IBM mainframe, they are part of the VMS operating system and are not in accordance with FORTRAN 77 standards. The VAX/VMS operating system does have a similar capability using the COMMAND or .COM file, however in an effort to make the programs more machine independent and compliant with the FORTRAN 77 standard, it was decided to open and define input and output files *within* each FORTRAN program.

The use of the OPEN statement causes a logical unit number (device) to be assigned for input and or output. Within the OPEN statement specific characteristics of the file such as record size, file type, type of access, file status, etc., are defined. An example of such an OPEN statement is shown in Figure 16.

```
C OPEN FILE FOR DATA FILE INPUT
      OPEN (UNIT=LUN,
     2      FILE= 'INFILE',
     2      ORGANIZATION= 'SEQUENTIAL',
     2      ACCESS= 'SEQUENTIAL',
     2      RECORDTYPE= 'VARIABLE',
     2      FORM= 'FORMATTED',
     2      STATUS= 'OLD')
C OPEN SCRATCH FILE FOR MATRIX INPUT TO SOLN ROUTINE
      OPEN (UNIT=2,
     2      FILE= 'JTFLAP2.DAT',
     2      ORGANIZATION= 'SEQUENTIAL',
     2      ACCESS= 'SEQUENTIAL',
     2      RECORDTYPE= 'VARIABLE',
     2      FORM= 'UNFORMATTED',
     2      STATUS= 'SCRATCH')
```

**Figure 16. Sample OPEN Statement**

Much of the information shown in these OPEN statements may be defaulted, that is, if a qualifier is not input by the programmer, a predetermined response is set by the compiler. The attributes have been shown here for clarity and to enhance portability. Since not all compilers use the same defaults, it is important to know as much as possible about the file attributes when transferring programs from one machine to another. General information on these qualifiers may be found in most FORTRAN texts and specifics for the MicroVAX 2000 may be found in the VAX FORTRAN Manuals [Refs. 16 and 17 ].

# VI. RESULTS AND RECOMMENDATIONS

The objectives of this thesis study have been achieved. A set of four FORTRAN programs for basic aerodynamic analysis are available for student projects on the MicroVAX 2000 CAD/CAE workstation. The following programs have been successfully transferred from the NPS IBM mainframe computer and are operational on the MicroVAX/2000.

- Program DUBLET
- Program PANEL
- Program VORLAT
- Program JETFLAP

In addition, an interactive program, JETFLAPIN, has been developed and implemented. The programs are easy to use, JETFLAP being an exception, and they provide the desired attributes of data review/correction, multiple run capability and error-checking. A users manual for each program was created. These manuals along with sample input/output files and complete program listings are contained in the appendices.

The programs were tested to ensure their accuracy and completeness following conversion. This was accomplished by comparing the output files generated by the IBM mainframe and the MicroVAX/2000 for identical input files. The numerical output values were generally in agreement to the fourth decimal place or better. When the JETFLAP output file for the DOUGLAS.DAT case[11] was compared to the output file in Ref. 7, it was found to be numerically exact, save for a few isolated values.

The results of the 2-D programs DUBLET and PANEL were compared to the expected theoretical values and wind tunnel data and showed good correlation. The results of program PANEL for the NASA LS(1)-0013 airfoil showed excellent agreement with those of Ref. 18. Although not its main purpose, the PANEL program is especially useful for generating the surface coordinates for an airfoil of the NACA XXXX or 23XXX series.

The 3-D program VORLAT, using the cosine spacing option, produced results nearly identical to those obtained by Hough [Ref. 19], for a wing of aspect ratio 2. As

---

11 Input file used by Douglas Aircraft Co. to validate JETFLAP in their report to ONR.

mentioned previously, the results of JETFLAP compared well with the results found in Refs. 7, 8 and 13.

Countless manhours were expended in the editing, debugging and validating of these programs, and the result is the desired set of baseline programs for basic aerodynamic class projects and research.

As with all programs, there are still a few more changes that could be made to improve the utility or flexibility of these programs. The next major step is to provide the capability of generating graphical output from the data produced by these programs. The programs DUBLET, PANEL and VORLAT lend themselves quite readily to this due to their columnar output form, and in fact, the results shown in Figures 26 through 34 in Appendix E were produced on the IBM mainframe using EASYPLOT and DISSPLA.

There is also further work to be done on program JETFLAPIN. Although it is fully operational, the data review/correction and error-trapping routines were not implemented for jet-flapped wings due to time constraints. A user inputing data for a conventional unblown wing of arbitrary or trapezoidal planform will not be aware of this deficiency.

Although the JETFLAPIN program performs its designed task of assisting the user in creating the properly formatted JETFLAP input file, a few suggestions for improvement are considered relevant.

- The program should allow the user to define the number of spanwise and chordwise divisions and then automatically compute the required coordinates using a semicircle or similar scheme.
- The program should provide graphical display of the spanwise and chordwise loadings for the fundamental and composite cases. The section loadings to be plotted should be user selectable.
- The capability to read in and either continue or modify an existing file would be quite useful. This would be an improvement over using the EDT editor to modify (and possibly corrupt) the properly formatted file.

# APPENDIX A. PROGRAM DUBLET USER'S MANUAL

## USERS GUIDE CONTENTS

**Introduction**

The purpose of the DUBLET program is to determine the piecewise constant doublet strength $m(t)$ for a line doublet distribution of an elliptic or airfoil-like shape at zero angle of attack. The points $t_i$ represent the location of the doublets along the chord or line of symmetry. They are concentrated near the ends of the distribution, using a cosine spacing method, where the variation of the doublet strength is expected to be most rapid. The point $t_1$ corresponds to $x_s$ and $t_N$ corresponds to the endpoint $x_f$. The abscissas $x_i$ of the points at which the integral equation is satisfied are chosen as the midpoints of the subintervals on which the doublet strength is constant, i.e., $x_i = (t_i + t_{i-1})/2$ .

The stream function can be calculated from the doublet strength distribution. From the stream function, the velocity components and the pressure coefficients may be calculated. The surface shape is defined by $y = Y(x)$ and the solution must satisfy the boundary conditions at the leading and trailing edge stagnation points.

**Assumptions and Limitations**

The approach taken to develop this method of solution assumes that the source and doublet strength functions are both piecewise-constant. It is also important to remember that this solution is for incompressible and inviscid irrotational flow. Since the bodies under investigation are symmetrical and at zero angle of attack, there is no lift or induced drag produced. In addition, there is no drag since we are considering an inviscid fluid.

**Input Description**

There are very few input values required for this simple program. Their description and program variable names are listed below.

**NTYPE** - Type of body shape; elliptic or airfoil-like.

**TAU** - Thickness ratio. (Maximum thickness/chord)

**XMAXY** - Chordwise location of the point of maximum thickness. (Airfoil only)

**N** - Number of intervals. $2 \leq N \leq 100$

**XS** - Doublet distribution starting point.

**XF** - Doublet distribution ending point.

**NXTOL** - Exponent value used to generate the convergence criterion **XTOL**.

**NFTOL** - Exponent value used to generate the convergence criterion **FTOL**.

**XTOL** - X location tolerance.

**FTOL** - X location tolerance.

**Sample Problem**

A few sample problems will illustrate the use of the DUBLET program. The first run will be done using an ellipse of thickness ratio 0.1. The second run will analyze an airfoil-like shape with a thickness ratio of 0.12 and a chordwise location of maximum thickness of 0.30.

**Starting the Program**

Begin with the screen showing the DCL prompt, which looks like this.

**$**

Next, ensure that the program is in your directory by typing

**DIR [Return]**

and viewing the files for DUBLET.EXE and DUBLET.OBJ. If only the DUBLET.FOR file exists, you must compile the program by typing,

**FOR DUBLET [Return]**

The next step is to link the program by entering,

**LINK DUBLET [Return]**

The files DUBLET.EXE and DUBLET.OBJ will now exist and you will be able to run the program.

**Running the Program**

To run the program, type

**DUBLET [Return]**

The program will start and the screen should look similar to what is shown in Figure 17.

```
PROGRAM DUBLET : VERSION 2 : 3 AUGUST 88

DOUBLET DISTRIBUTION METHOD IS USED TO DETERMINE
INCOMPRESSIBLE FLOW AROUND AN ELLIPSE OR
SYMMETRICAL AIRFOIL AT ZERO ANGLE OF ATTACK

PROGRAM ASSUMES A NONDIMENSIONAL CHORD, THAT IS,
THE VALID RANGE OF X IS FROM 0 TO 1.

ENTER TYPE OF BODY SHAPE DESIRED:
     1) ELLIPTIC OR
     2) SYMMETRICAL AIRFOIL-LIKE
ENTER 1 OR 2.
```

**Figure 17. Initial Screen for Program DUBLET**

For the elliptic case respond to the request by entering

**1 [Return]**

Respond to the request for the thickness ratio by entering

**0.1 [Return]**

Now enter the number of intervals you desire the doublet distribution to have by entering

**10 [Return]**

The screen should now look like what is shown in Figure 18.

```
WHICH METHOD DO YOU WISH TO USE TO DETERMINE THE
DOUBLET DISTRIBUTION ENDPOINTS? (1 OR 2)
  1) PROGRAM INTERVAL-HALVING SUBROUTINE TO ITERATE.
  2) MANUAL ITERATION BY THE USER.
```

**Figure 18. Endpoint Determination Method Selection Screen**

Respond to the question by entering

**1 [Return]**

If you should desire to enter your own values, enter **2**.

The next values you will be required to enter are for the X location tolerance and the stagnation point velocity function tolerance. It is recommended that values of 10E-6 (0.000001) be used. The maximum number of iterations should be set at a value of at least 20 when using such small tolerances.

The output parameter entry has only to do with the interval halving subroutine. Unless you are having problems with the program or are interested in the convergence of the solution, it is recommended that this value be set to zero (0).

Following entry of the output parameter, the program begins the solution process. It returns with U0 and U1, the values for the X velocity component at the stagnation points and the values for XS and XF, the beginning and ending points of the line doublet distribution. If the values for U0 and U1 are sufficiently close to zero, say less than 10E-3 (0.001), then enter

**Y [Return]**

If you desire more accuracy, enter

**N [Return]**

and then reenter the tolerance and maximum iteration values. Responding with a (Y) will cause the program to proceed to the output stage. Values will be printed to the screen and to the following data files:

```
DUBLET.DAT   :  DOUBLET STRENGTH DISTRIBUTION
SHAPE.DAT    :  BODY SURFACE COORDINATES
PRESSURE.DAT:   SURFACE PRESSURE DISTRIBUTION
```

You will be asked for the number of pressure coefficient output points you desire. This number is independent of the number of intervals of the line doublet distribution. It affects only the number of output data points and not the accuracy of the solution. The program now asks if you want to make another run. Enter

**1 [Return]**

This time the sample problem will work through the airfoil-like shape case and the user will supply the values of XS and XF. The user may experiment with manual iteration, however to save space this sample will use previously determined satisfactory values of XS and XF for the initial guess.

You should now be back at the initial screen and it should look like Figure 17. For the airfoil-like case enter

**2 [Return]**

Respond to the request for the thickness ratio by entering

**.12 [Return]**

For the chordwise location of maximum thickness, enter

**.30 [Return]**

Now enter the number of intervals you desire the doublet distribution to have by entering

**10 [Return]**

The next step is to select the method for the determination of the endpoints for the doublet distribution. The screen should look like Figure 19.

```
WHICH METHOD DO YOU WISH TO USE TO DETERMINE THE
DOUBLET DISTRIBUTION ENDPOINTS? (1 OR 2)
  1) PROGRAM INTERVAL-HALVING SUBROUTINE TO ITERATE.
  2) MANUAL ITERATION BY THE USER.
```

**Figure 19. Endpoint Determination Method Selection Screen**

This time respond to the question by entering

**2 [Return]**

For the doublet distribution starting point, XS, enter

**.0082129128 [Return]**

For the doublet distribution ending point, XF, enter

**.9994138 [Return]**

As with the previous example, the program now begins the solution process. It returns with U0 and U1, the values for the X velocity component at the stagnation points. It also echoes back the values entered for XS and XF. If the returned values for U0 and U1 are sufficiently close to zero, then enter

**Y [Return]**

This response will cause the program to proceed to the output stage. Values will be printed to the screen and to the data files.

Enter the number of pressure coefficient output points you desire. You are reminded that this number is independent of the number of intervals of the line doublet distribution and it does not affect the accuracy of the solution.

The program now asks if you want to make another run. The session is finished, so enter

**2 [Return]**

This completes the sample problems for the DUBLET program. The data files created by these sample runs and the listing for the DUBLET program are on the following pages. Since the bodies analyzed by this program are symmetrical with respect to the x axis, only the upper surface body shape coordinates and pressure coefficients are output. For this reason, the piecewise constant doublet strength M(I) is divided by two to indicate the portion affecting the upper surface.

## SAMPLE PROBLEM OUTPUT DATA FILES

**Sample problem 1: Ellipse - Thickness ratio = 0.1**

**T(I) = Chordwise location of doublets, T(1) = XS T(N) = XF**
**M(I)/2 = Piecewise doublet strength / 2**

DATA FILE: DUBLET.DAT

DOUBLET STRENGTH DISTRIBUTION

| T(I) | M(I)/2 |
|---|---|
| 0.0045 | 0.0112 |
| 0.0287 | 0.0259 |
| 0.0991 | 0.0395 |
| 0.2087 | 0.0494 |
| 0.3469 | 0.0547 |
| 0.5000 | 0.0547 |
| 0.6531 | 0.049[illegible] |
| 0.7913 | 0.0395 |
| 0.9009 | 0.0259 |
| 0.9713 | 0.0112 |
| 0.9955 | 0.0000 |

**Sample problem 2: Airfoil Shape - Thickness ratio = 0.12, XMAXY = 0.30**

DOUBLET STRENGTH DISTRIBUTION

| T(I) | M(I)/2 |
|---|---|
| 0.0082 | 0.0184 |
| 0.0325 | 0.0438 |
| 0.1029 | 0.0624 |
| 0.2125 | 0.0703 |
| 0.3507 | 0.0671 |
| 0.5038 | 0.0551 |
| 0.6570 | 0.0383 |
| 0.7951 | 0.0214 |
| 0.9048 | 0.0083 |
| 0.9752 | 0.0016 |
| 0.9994 | 0.0000 |

**Sample problem 1: Ellipse - Thickness ratio = 0.1**

DATA FILE: SHAPE.DAT

BODY SHAPE - UPPER SURFACE

| X | Y |
|---|---|
| 0.0166 | 0.0128 |
| 0.0639 | 0.0245 |
| 0.1539 | 0.0361 |
| 0.2778 | 0.0448 |
| 0.4234 | 0.0494 |
| 0.5766 | 0.0494 |
| 0.7222 | 0.0448 |
| 0.8461 | 0.0361 |
| 0.9361 | 0.0245 |
| 0.9834 | 0.0128 |

**Sample problem 2: Airfoil Shape - Thickness ratio = 0.12, XMAXY = 0.30**

BODY SHAPE - UPPER SURFACE

| X | Y |
|---|---|
| 0.0203 | 0.0219 |
| 0.0677 | 0.038[illegible] |
| 0.1577 | 0.0523 |
| 0.2816 | 0.0597 |
| 0.4272 | 0.0586 |
| 0.5804 | 0.0500 |
| 0.7260 | 0.0365 |
| 0.8499 | 0.0216 |
| 0.9400 | 0.0091 |
| 0.9873 | 0.0020 |

**Sample problem 1: Ellipse - Thickness ratio = 0.1**

DATA FILE: PRESSURE.DAT

BODY SURFACE PRESSURE DISTRIBUTION

| X | CP |
|---|---|
| 0.0000 | 1.0000 |
| 0.1111 | -0.2621 |
| 0.2222 | -0.2341 |
| 0.3333 | -0.1866 |
| 0.4444 | -0.2078 |
| 0.5556 | -0.2078 |
| 0.6667 | -0.1866 |
| 0.7778 | -0.2341 |
| 0.8889 | -0.2621 |
| 1.0000 | 1.0000 |

**Sample problem 2: Airfoil Shape - Thickness ratio = 0.12, XMAXY = 0.30**

BODY SURFACE PRESSURE DISTRIBUTION

| X | CP |
|---|---|
| 0.0000 | 1.0000 |
| 0.1111 | -0.3946 |
| 0.2222 | -0.3572 |
| 0.3333 | -0.3162 |
| 0.4444 | -0.2938 |
| 0.5556 | -0.1820 |
| 0.6667 | -0.1180 |
| 0.7778 | -0.2180 |
| 0.8889 | -0.2142 |
| 1.0000 | 1.0000 |

## PROGRAM DUBLET LISTING

```
      PROGRAM DUBLET
C
C *** MODIFIED FOR USE ON THE MICROVAX/2000 BY J.A. CAMPBELL (JUL 88)
C     FINAL UPDATES MADE 14 SEP 88 - (JAC)
C ****************************************************************
C
C         INCOMPRESSIBLE AERODYNAMICS OF SYMMETRIC AIRFOIL
C         AT ZERO ANGLE OF ATTACK BY LINE DOUBLET DISTRIBUTION
C
C     ORIGINAL IBM MAINFRAME PROGRAM WAS ADAPTED FROM JACK MORAN'S BOOK
C     'AN INTRODUCTION TO THEORETICAL AND COMPUTATIONAL AERODYNAMICS',
C     WILEY AND SONS, NEW YORK 1984.  THE LISTING IS FOUND ON PAGE 75.
C
C     PROGRAM FLEXIBILITY AND USER INTERFACE WAS REVISED FOR
C     PROFESSOR J.V. HEALEY BY JOHN CAMPBELL.  JULY 1988.
C
C ****************************************************************
C
      CHARACTER*1  IANS
      INTEGER    NANS
      COMMON   T(100),M(100),N,XS,XF
      COMMON /FCN/ AX,TAU,NTYPE
      REAL     M,MPLOT
C
C FOLLOWING LINES FOR OUTPUT FILES ADDED BY J.A. CAMPBELL (AUG88)
C    OPEN FILE FOR DOUBLET STRENGTH DISTRIBUTION OUTPUT
      OPEN (UNIT=11,
     2        FILE= 'DUBLET.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'UNKNOWN')
C
C    OPEN FILE FOR BODY SHAPE OUTPUT
      OPEN (UNIT=12,
     2        FILE= 'SHAPE.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'UNKNOWN')
C
C    OPEN FILE FOR BODY SURFACE PRESSURE DISTRIBUTION OUTPUT
      OPEN (UNIT=13,
     2        FILE= 'PRESSURE.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'UNKNOWN')
C
C
C   CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THE PRINT HEADER
    5 CONTINUE
      CALL CLRSCRN
      PRINT *
      PRINT *, ' PROGRAM DUBLET : VERSION 2 : 9 SEPTEMBER 88 '
      PRINT *
      PRINT *, ' DOUBLET DISTRIBUITON METHOD IS USED TO DETERMINE'
      PRINT *, ' INCOMPRESSIBLE AERODYNAMICS OF AN ELLIPSE OR '
      PRINT *, ' SYMMETRICAL AIRFOIL AT ZERO ANGLE OF ATTACK'
      PRINT *
      PRINT *, ' PROGRAM ASSUMES A NONDIMENSIONAL CHORD, THAT IS,'
      PRINT *, ' THE VALID RANGE OF X IS FROM 0 TO 1.'
      PRINT *
   10 PRINT *, ' ENTER TYPE OF BODY SHAPE DESIRED:      '
      PRINT *, '        1) ELLIPTIC OR'
      PRINT *, '        2) SYMMETRICAL AIRFOIL-LIKE'
      PRINT *, ' ENTER 1 OR 2.'
   15 READ (5,*)  NTYPE
      IF (NTYPE .LT. 1 .OR. NTYPE .GT. 2)  THEN
         PRINT *, '   INVALID ENTRY. ENTER 1 OR 2.'
         GO TO 15
      END IF
      PRINT *,'   ENTER THICKNESS RATIO (TAU).'
      READ (5,*) TAU
      IF (NTYPE .GT. 1) THEN
         PRINT *
         PRINT *,'   ENTER THE NONDIMENSIONAL X LOCATION OF MAXIMUM',
     +   ' THICKNESS.'
   20    READ (5,*) XMAXY
      IF (XMAXY .GT. 0.5) THEN
         PRINT *,'   THE PROGRAM CONSIDERS THE ONSET FLOW TO BE'
         PRINT *,'   APPROACHING FROM THE LEFT.  THEREFORE, THE'
         PRINT *,'   X LOCATION OF MAXIMUM THICKNESS MUST BE < 0.5.'
         PRINT *,'   ==>  PLEASE REENTER.'
         GO TO 20
      END IF
         AX = (.5 * TAU)/(SQRT(XMAXY)*(1. - XMAXY))
      END IF
C
C        INPUT NUMBER OF INTERVALS N
C
      PRINT *
   70 PRINT *,  ' ENTER NUMBER OF INTERVALS DESIRED. N ='
   71 READ (5,*)    N
```

```
      PRINT *
      IF(N .LT. 2 .OR. N .GT. 100) THEN
        WRITE(6,21) N
        PRINT *, '     A MINIMUM OF TWO INTERVALS AND A MAXIMUM OF'
        PRINT *, '     100 IS ALLOWED. ===>  PLEASE REENTER.'
        GO TO 71
      END IF
   21 FORMAT(1X,5X,'NUMBER OF INTERVALS REQUESTED =',I3)
C
C   ASK USER FOR AUTOMATIC OR MANUAL DETERMINATION OF ENDPOINTS.
   80 CONTINUE
      CALL CLRSCRN
      PRINT *
      PRINT *, ' WHICH METHOD DO YOU WISH TO USE TO DETERMINE THE'
      PRINT *, ' DOUBLET DISTRIBUTION ENDPOINTS? (1 OR 2)'
      PRINT *, '     1) PROGRAM INTERVAL HALVING SUBROUTINE TO ITERATE.'
      PRINT *, '     2) MANUAL ITERATION BY THE USER.'
      PRINT *
      PRINT *, '          ENTER 3 TO RETURN TO START.'
      READ (5,*) NMETH
      GO TO (120,100,5) NMETH
C
C   MANUALLY DETERMINE ENDPOINTS OF SOURCE DISTRIBUTION XS, XF
C
  100 CONTINUE
      CALL CLRSCRN
      PRINT *
      PRINT *, '       ROUTINE FOR MANUAL DETERMINATION OF ENDPOINTS'
      PRINT *
      PRINT *, '     --------------------------------------------'
      PRINT *
      PRINT *, ' ENTER THE DOUBLET DISTRIBUTION STARTING POINT, XS.'
      PRINT *, ' (XS SHOULD BE APPROXIMATELY ONE HALF OF'
      PRINT *, ' THE NONDIMENSIONAL LEADING EDGE RADIUS.)'
      READ (5,*) XS
      PRINT *
      PRINT *, ' ENTER THE DOUBLET DISTRIBUTION ENDING POINT, XF.'
      PRINT *, ' (XF SHOULD BE APPROXIMATELY ONE MINUS HALF'
      PRINT *, ' OF THE NONDIMENSIONAL TRAILING EDGE RADIUS.)'
      READ (5,*) XF
      PRINT *
      PRINT *
      CALL FINDM (T,M,N,XS,XF)
      CALL PRESS(0.0,U0,CP0)
      CALL PRESS(1.0,U1,CP1)
      GO TO 150
C
  120 CONTINUE
      CALL CLRSCRN
      PRINT *
      PRINT *, '       INTERVAL HALVING ROUTINE FOR DETERMINATION OF'
      PRINT *, '              DOUBLET DISTRIBUTION ENDPOINTS'
      PRINT *
      PRINT *, '     --------------------------------------------'
      PRINT *
C       ENTER THE PARAMETERS REQUIRED BY THE INTERVAL HALVING METHOD
C       WHICH IS USED TO OBTAIN THE PROPER LOCATIONS FOR XS AND XF.
      PRINT *, ' ENTER THE INTEGER EXPONENT FOR THE X TOLERANCE, NXTOL.'
      PRINT *, '    EXAMPLE:  A VALUE OF 4, GIVES A TOLERANCE OF 0.0001.'
      READ (5,*) NXTOL
      PRINT *
      PRINT *, ' ENTER THE INTEGER EXPONENT FOR THE FUNCTION ',
     &'TOLERANCE, NFTOL.'
      PRINT *, ' (SAME IDEA AS NXTOL; 5 YIELDS FTOL = 0.00001).'
      READ (5,*) NFTOL
      PRINT *
      PRINT *, ' ENTER THE MAXIMUM NUMBER OF ITERATIONS, MAXIT, TO '
      PRINT *, ' LOCATE XS AND XF. (FOR NFTOL = 6, SUGGEST 35-40)'
      READ (5,*) MAXIT
      PRINT *
      PRINT *, ' ENTER THE OUTPUT PARAMETER, IOUT.'
      PRINT *, '      IOUT = 0 TO SUPPRESS ALL ITERATION RELATED OUTPUT'
      PRINT *, '             1 TO OUTPUT FINAL RESULTS ONLY'
      PRINT *, '             2 TO OUTPUT DETAILS FOR EACH ITERATION'
      READ (5,*) IOUT
      CALL INTHV (NXTOL,NFTOL,NTYPE,MAXIT,IOUT,U0,U1)
C  RUN THROUGH PROCESS AGAIN WITH FINAL VALUES OBTAINED BY ITERATION
      CALL FINDM (T,M,N,XS,XF)
      CALL PRESS(0.0,U0,CP0)
      CALL PRESS(1.0,U1,CP1)
C
  150 PRINT *, ' U AT X = 0 =',U0,'      XS =',XS
      PRINT *, ' U AT X = 1 =',U1,'      XF =',XF
      PRINT *
      PRINT *, ' THESE VALUES FOR U SHOULD BE NEAR ZERO.'
      PRINT *, ' DO YOU ACCEPT THESE RESULTS (Y/N)'
      READ 1000, IANS
      IF (IANS .NE. 'Y') THEN
        GO TO (120,100) NMETH
      END IF
C
C            OUTPUT RESULTS
C
      PRINT 1010
      WRITE (11,1012)
      M(N+1) = 0.0
      DO 200  I = 1,N+1
      MPLOT   = M(I)*3.1415926585
      PRINT 1040, T(I),MPLOT
  200 WRITE (11,1040) T(I),MPLOT
      PRINT 1020
```

```
      WRITE (12,1020)
      DO 210  I = 1,N
      XX        = .5*(T(I) + T(I+1))
      YY        = Y(XX)
      PRINT 1040, XX,YY
  210 WRITE (12,1040) XX,YY
      PRINT 1030
  212 READ (5,*)    NPRINT
      IF (NPRINT .LT. 2) THEN
         PRINT *, '  YOU MUST ENTER A MINIMUM OF 2. PLEASE REENTER.'
         GO TO 212
      END IF
      WRITE (13,1032)
      DO 220  I = 1,NPRINT
      XX        = (I-1)/FLOAT(NPRINT-1)
      CALL PRESS(XX,U,CP)
      PRINT 1040, XX,CP
  220 WRITE (13,1040) XX,CP
C  CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THEN PRINT HEADER
      CALL CLRSCRN
      PRINT *
      PRINT *, ' PROGRAM DUBLET RESULTS HAVE BEEN WRITTEN TO FILES:'
      PRINT *
      PRINT *, ' DUBLET.DAT   :  DOUBLET STRENGTH DISTRIBUTION'
      PRINT *, ' SHAPE.DAT    :  BODY SURFACE COORDINATES'
      PRINT *, ' PRESSURE.DAT:  SURFACE PRESSURE DISTRIBUTION'
      PRINT *
      PRINT *
C        OPTION TO MAKE ANOTHER RUN
      PRINT *
      PRINT *, ' DO YOU WISH TO:    '
      PRINT *, '        1) MAKE ANOTHER RUN OR'
      PRINT *, '        2) END THIS SESSION'
      PRINT *, ' ENTER 1 OR 2.'
      PRINT *
      CALL QUERY (NANS)
      CALL CLRSCRN
      IF (NANS .EQ. 1)  GO TO 10
      STOP
 1000 FORMAT(A1)
 1010 FORMAT(/,' DOUBLET STRENGTH DISTRIBUTION',/,
     +       ' M = M(I) FOR T(I) .LT. T .LT. T(I+1)',//,
     +       5X,'T(I)',5X,'M(I)/2',/)
 1012 FORMAT(/,' DOUBLET STRENGTH DISTRIBUTION',//,
     +       5X,'T(I)',5X,'M(I)/2',/)
 1020 FORMAT(//,' BODY SHAPE - UPPER SURFACE',//,6X,'X',9X,'Y',/)
 1030 FORMAT(//,' BODY SURFACE PRESSURE DISTRIBUTION',//,
     +       6X,'X',8X,'CP',//,' INPUT NUMBER OF PRESSURE COEFFICIENT',
     +       ' OUTPUT POINTS')
 1032 FORMAT(//,' BODY SURFACE PRESSURE DISTRIBUTION',//,
     +       6X,'X',8X,'CP',//)
 1040 FORMAT(2F10.4)
      END
C***************************************************************
      SUBROUTINE CLRSCRN
C
C  LIBRARY ROUTINE TO CLEAR THE SCREEN.
C
      ISTAT = LIB$ERASE_PAGE (1,1)
      RETURN
C***************************************************************
      END
      SUBROUTINE QUERY(NANS)
C
C  ROUTINE TO TRAP ERRORS CAUSED BY IMPROPER RESPONSES TO QUESTIONS.
C  THE COMPUTER GENERATES AND ERROR WHEN A CHARACTER IS SUPPLIED TO
C  A QUESTION EXPECTING AN INTEGER OR REAL VALUE.
C
      NOTEST=0
    1 CONTINUE
      IF (NOTEST .GT. 0) THEN
         PRINT *, '  CHARACTER VALUES ARE NOT VALID.'
         PRINT *, '  PLEASE ENTER A VALUE OF 1 OR 2.'
      END IF
      NOTEST = NOTEST + 1
      READ (5,*,ERR=1)NANS
      RETURN
C***************************************************************
      END

      SUBROUTINE FINDM (T,M,N,XS,XF)
C
C             FIND DOUBLET STRENGTH TO MEET
C             FLOW TANGENCY CONDITION
C
      DIMENSION  T(100),M(100)
      COMMON /COF/ A(101,111),NEQNS
      REAL M
      PI        = 3.1415926585
      NP        = N + 1
      DO 100  I = 1,NP
C     COSINE SPACING SCHEME FROM XS TO XF
      FRACT     = .5*(1. - COS(PI*(I-1)/FLOAT(N)))
  100 T(I)      = XS + (XF - XS)*FRACT
C
C             SET UP LINEAR SYSTEM OF EQUATIONS
C
      DO 210  I = 1,N
      XI        = .5*(T(I) + T(I+1))
      YI        = Y(XI)
      FAC1      = ATAN2(T(1) - XI,YI)
```

```
      DO 200  J = 1,N
      FAC2    = ATAN2(T(J+1) - XI,YI)
      A(I,J)  = (FAC2 - FAC1)/YI
  200 FAC1    = FAC2
  210 A(I,NP) = 1.0
C
C             SOLVE FOR DOUBLET STRENGTH
C
      NEQNS   = N
      CALL GAUSS(1)
      DO 300  I = 1,N
  300 M(I)    = A(I,NP)
      RETURN
C**********************************************************************
      END
      SUBROUTINE GAUSS(NRHS)
C
C        SOLUTION OF LINEAR ALGEBRAIC SYSTEM BY
C        GAUSS ELIMINATION WITH PARTIAL PIVOTING
C
C             °A          = COEFFICIENT MATRIX
C             NEQNS       = NUMBER OF EQUATIONS
C             NRHS        = NUMBER OF RIGHT HAND SIDES
C
C             RIGHT-HAND SIDES AND SOLUTIONS STORED IN
C             COLUMNS NEQNS+1 THRU NEQNS+NRHS OF °A
C
      COMMON /COF/ A(101,111),NEQNS
      NP      = NEQNS + 1
      NTOT    = NEQNS + NRHS
C
C             GAUSS REDUCTION
C
      DO 150  I = 2,NEQNS
C
C             --  SEARCH FOR LARGEST ENTRY IN (I-1)TH COLUMN
C                 ON OR BELOW MAIN DIAGONAL
C
      IM      = I - 1
      IMAX    = IM
      AMAX    = ABS(A(IM,IM))
      DO 110  J = I,NEQNS
      IF (AMAX .GE. ABS(A(J,IM))) GO TO 110
      IMAX    = J
      AMAX    = ABS(A(J,IM))
  110 CONTINUE
C
C             --   SWITCH (I-1)TH AND IMAXTH EQUATIONS
C
      IF (IMAX .NE. IM)   GO TO 140
      DO 130  J = IM,NTOT
      TEMP    = A(IM,J)
      A(IM,J) = A(IMAX,J)
      A(IMAX,J) = TEMP
  130 CONTINUE
C
C             ELIMINATE (I-1)TH UNKNOWN FROM
C             ITH THRU (NEQNS)TH EQUATIONS
C
  140 DO 150  J = I,NEQNS
      R       = A(J,IM)/A(IM,IM)
      DO 150  K = I,NTOT
  150 A(J,K)  = A(J,K) - R*A(IM,K)
C
C             BACK SUBSTITUTION
C
      DO 220  K = NP,NTOT
      A(NEQNS,K) = A(NEQNS,K)/A(NEQNS,NEQNS)
      DO 210  L = 2,NEQNS
      I       = NEQNS + 1 - L
      IP      = I + 1
      DO 200  J = IP,NEQNS
  200 A(I,K)  = A(I,K) - A(I,J)*A(J,K)
  210 A(I,K)  = A(I,K)/A(I,I)
  220 CONTINUE
      RETURN
C**********************************************************************
      END
      SUBROUTINE PRESS(X,U,CP)
C
C             FIND PRESSURE COEFFICIENT CP AT (X,Y(X))
C
      COMMON  T(100),M(100),N,XS,XF
      REAL    M
      YB      = Y(X)
      U       = 1.0
      V       = 0.0
      VF1     = 1./((T(1) - X)**2 + YB*YB)
      UF1     = (T(1) - X)*VF1
      DO 100  J = 1,N
      VF2     = 1./((T(J+1) - X)**2 + YB*YB)
      UF2     = (T(J+1) - X)*VF2
      U       = U + M(J)*(UF2 - UF1)
      V       = V - M(J)*YB*(VF2 - VF1)
      VF1     = VF2
  100 UF1     = UF2
      CP      = 1.0 - U*U - V*V
      RETURN
      END
```

```
      FUNCTION Y(X)
      COMMON /FCN/ AX,TAU,NTYPE
C
C        ORDINATE OF BODY CONTOUR
C
      IF (NTYPE .EQ. 1) THEN
C
C        PROVIDE BODY ORDINATES FOR AN ELLIPSE OF THICKNESS RATIO TAU
C        (CHORD HAS BEEN NONDIMENSIONALIZED, C=1.0)
C
C        TO REDUCE THE NUMBER OF VARIABLES PASSED IN THE FUNCITON
C        STATEMENT, THE DUMMY VARIABLE AX PASSES TAU FOR THE ELLIPSOID
C        CASE AND THE COEFFICIENT AX(TAU,XMAXY) FOR THE SYMMETRICAL
C        AIRFOIL-LIKE CASE.
C
      Y        = TAU * SQRT(X*(1.-X))
         ELSE
C
C        PROVIDE BODY ORDINATES FOR A SYMMETRIC AIRFOIL-LIKE SHAPE
C        (CHORD HAS BEEN NONDIMENSIONALIZED, C=1.0)
C
      Y        =  AX * SQRT(X)*(1-X)
      END IF
      RETURN
C*********************************************************************
      END
C
      SUBROUTINE INTHV (NXTOL,NFTOL,NTYPE,MAXIT,IOUT,U0,U1)
      COMMON  T(100),M(1 )),N,XS,XF
C    SUBROUTINE TO FIND THE ROOTS OF f(x) = 0 USING THE
C    INTERVAL HALVING METHOD
C
C    IN THE PARAMETER LIST THE USER MUST PROVIDE:
C          NXTOL = EXPONENT FOR X TOLERANCE VALUE
C          NFTOL = EXPONENT FOR FUNCTION TOLERANCE VALUE
C          NTYPE = SHAPE TYPE; ELLIPTICAL OR AIRFOIL
C          MAXIT = MAXIMUM NUMBER OF ITERATIONS
C           IOUT = 0 TO SUPPRESS ALL OUTPUT (TO DEVICE IW)
C                  1 TO OUTPUT FINAL RESULTS ONLY
C                  2 TO OUTPUT DETAILS FOR EACH ITERATION
C    THE SUBROUTINE CALCULATES:
C       XPREV, X = TWO INITIAL GUESSES, GIVEN N
C    THE SUBROUTINE RETURNS:
C          XS, XF = CURRENT X VALUES WHEN TERMINATION OCCURRED
C          U0, U1 = CURRENT VELOCITY VALUES WHEN TERMINATION OCCURRED
C           IEXIT = 1, 2, 3, 4 OR 7 (SEE FORMAT STATEMENTS 1 - 4 & 7)
C    Subprogram name F must be declared EXTERNAL in calling program.
C
C
      IW = 5
      XTOL = 10.**(-NXTOL)
      FTOL = 10.**(-NFTOL)
C    CALCULATE INITIAL GUESS FOR XS AND XF, GIVEN N
      XS = 1. / FLOAT(N + 1)
      XSPREV = 10.**(-6)
      XF = 1. - XS
      XFPREV = 1. - XSPREV
C    SET X VALUES FOR LEADING AND TRAILING EDGES FOR SUBROUTINE PRESS
      XLE = 0.0
      XTE = 1.0
C
C    ITERATE TO DETERMINE THE PROPER LOCATION FOR XF
C
C    FIRST CHECK TO SEE THAT F(XF) & F(XFPREV) DIFFER IN SIGN
C    SO THAT THE METHOD WILL CONVERGE.
C
C    EVALUATE PREVIOUS X VALUE
      CALL FINDM (T,M,N,XS,XFPREV)
      CALL PRESS (XTE,U1,CP)
      YFPREV = U1
C    EVALUATE INITAL GUESS FOR X VALUE
      CALL FINDM (T,M,N,XS,XF)
      CALL PRESS (XTE,U1,CP)
      YF = U1
      IF (IOUT .GT. 1) WRITE (IW,5) XFPREV, YPREV, XF, YF
      IF (YFPREV*YF .GT. 0.0) THEN
         I = -2
         PRINT 201
         RETURN
      END IF
C
C    COMPUTE SEQUENCE OF POINTS CONVERGING TO THE ROOT
      IEXIT = 1
      DO 10 K=1, MAXIT
         XR = (XFPREV + XF)/2.0
C    FOR THE ELLIPTIC CASE XS AND XF WILL BE EQUIDISTANT FROM THE EDGES.
         IF (NTYPE .LT. 2) THEN
            XS = ABS (1. - XR)
         END IF
         CALL FINDM (T,M,N,XS,XR)
         CALL PRESS (XTE,U1,CP)
         Y = U1
C    CHECK ON STOPPING CRITERIA
C
         DELTAXF = XFPREV-XR
         XERR = ABS(XFPREV-XR)/2.0
         IF (IOUT .GT. 1) WRITE (IW,6) K,XR,Y,DELTAXF
         IF (Y .EQ. 0.0) IEXIT = 2
         IF (ABS(Y) .LE. FTOL) IEXIT = 3
         IF ( XERR .LE. XTOL) IEXIT = 7
         IF (IEXIT .GT. 1) GO TO 20
```

```
         IF (Y*YFPREV .GT. 0.0) THEN
            XFPREV = XR
            YFPREV = Y
         ELSE
            XF = XR
            YF = Y
         END IF
   10 CONTINUE
C     THE MAXIMUM ITERATIONS HAS BEEN EXCEEDED,WITHOUT FINDING A ROOT.
      IEXIT = 4
   20 IF (IOUT .EQ. 0) GO TO 30
      IF (IEXIT .EQ. 1 ) WRITE (IW, 1) XR
      IF (IEXIT .EQ. 2 ) WRITE (IW, 2) XR
      IF (IEXIT .EQ. 3 ) WRITE (IW, 3) XR, NUMSIG
      IF (IEXIT .EQ. 4 ) WRITE (IW, 4) MAXIT
   30 CONTINUE
C     FOR THE ELLIPTIC CASE XS ANND XF ARE DETERMINED, SO GO BACK.
C
      IF (NTYPE .LT. 2) THEN
         CALL FINDM (T,M,N,XS,XF)
         CALL PRESS (XLE,U0,CP)
         GO TO 90
      END IF
C     NOW DO THE SAME FOR XS
      PRINT *, '    VALUE OBTAINED FOR XF',XF
      PRINT *, '    -- WORKING ON XS.'
C     EVALUATE PREVIOUS X VALUE
      CALL FINDM (T,M,N,XSPREV,XF)
      CALL PRESS (XLE,U0,CP)
      YSPREV = U0
C     EVALUATE INITAL GUESS FOR X VALUE
      CALL FINDM (T,M,N,XS,XF)
      CALL PRESS (XLE,U0,CP)
      YS = U0
      IF (IOUT .GT. 1) WRITE (IW,5) XSPREV, YSPREV, XS, YS
      IF (YSPREV*YS .GT. 0.0) THEN
         I = -2
         PRINT 201
         RETURN
      END IF
C
C     COMPUTE SEQUENCE OF POINTS CONVERGING TO THE ROOT
      IEXIT = 1
      DO 40 K=1, MAXIT
         XR = (XSPREV + XS)/2.0
         CALL FINDM (T,M,N,XR,XF)
         CALL PRESS (XLE,U0,CP)
         Y = U0
C     CHECK ON STOPPING CRITERIA
C
         DELTAXS = XSPREV-XR
         XERR = ABS(XSPREV-XR)/2.0
         IF (IOUT .GT. 1) WRITE (IW,6) K,XR,Y,DELTAXS
         IF (Y .EQ. 0.0) IEXIT = 2
         IF (ABS(Y) .LE. FTOL) IEXIT = 3
         IF ( XERR .LE. XTOL) IEXIT = 7
         IF (IEXIT .GT. 1) GO TO 50
         IF (Y*YSPREV .GT. 0.0) THEN
            XSPREV = XR
            YSPREV = Y
         ELSE
            XS = XR
            YS = Y
         END IF
   40 CONTINUE
C     THE MAXIMUM ITERATIONS HAS BEEN EXCEEDED,WITHOUT FINDING A ROOT.
      IEXIT = 4
   50 IF (IOUT .EQ. 0) RETURN
      IF (IEXIT .EQ. 1 ) WRITE (IW, 1) XR
      IF (IEXIT .EQ. 2 ) WRITE (IW, 2) XR
      IF (IEXIT .EQ. 3 ) WRITE (IW, 3) XR, NUMSIG
      IF (IEXIT .EQ. 4 ) WRITE (IW, 4) MAXIT
      IF (IEXIT .EQ. 7 ) WRITE (IW, 7) XR, XTOL
   90 RETURN
C***************************************************************
C
C
C     THIS SHOULD RETURN WITH U0 NEAR ZERO AND A GOOD VALUE OF XS.
    1 FORMAT('0SLOPE = 0 WHEN X =',G12.7,'.  ITERATION DISCONTINUED.')
    2 FORMAT('0COMPUTED F( ',G12.7,') IS 0.  ITERATION DISCONTINUED.')
    3 FORMAT('0ROOT:    ',G12.7,' APPEARS TO BE ACCURATE TO ',I1,'S.')
    4 FORMAT('0DESIRED ACCURACY IS NOT EVIDENT IN ',I3,' ITERATIONS.')
    5 FORMAT('0HALVING METHOD:  X(-1), X(0) ARE INITIAL GUESSES.',/,
     &       '0 K',4X,'X = XK',7X,'Y = F(X)',7X,'X-XPREV',/,
     &       ' -1   ', G12.7, E12.5, /,'  0   ', G12.7, E12.5)
    6 FORMAT(I3, 3X, G12.7, E12.5, E15.5)
    7 FORMAT('0X LOCATION: ',G12.7,' IS WITHIN X TOLERANCE OF ',E12.5)
  201 FORMAT('0FUNCTION HAS THE SAME SIGN AT BOTH INITIAL POSITIONS.'
     &       ,/,'0THE BUILT-IN ITERATION SCHEME WILL NOT WORK, THEREFORE'
     &       ,/,'0YOU MUST SELECT THE ENDPOINTS MANUALLY.')
      END
```

# APPENDIX B. PROGRAM PANEL USER'S MANUAL

## USERS GUIDE CONTENTS

## Introduction

The purpose of the PANEL program is to provide an analysis of the aerodynamics of NACA four-digit airfoils and airfoils of the NACA 230XX family using the panel method. This program has been modified to accept arbitrary airfoil surface coordinate input.

## Assumptions and Limitations

This program is limited to single-element airfoils. The solution is determined for conditions of incompressible and inviscid irrotational flow. Since we are considering an inviscid fluid, the coefficient of drag provided in the results is for the induced drag component only.

## Input Description

As with the DUBLET program, there are very few input values required for this simple program. Their description and program variable names are listed below.

**NUPPER** - Number of nodes on the upper surface.

**NLOWER** - Number of nodes on the lower surface.

**X(I),Y(I)** - Surface coordinates. These may be entered from the keyboard, from a data file, or from data statements. The program is capable of generating an approximation for airfoils of the NACA XXXX and 230XX series.

**ALPHA** - Angle of attack. (Angle between the chord and the freestream velocity.)

## Input Restrictions

The program, as written, is limited to 100 total surface nodes. This may be modified by changing the size of the arrays, however only a very complex surface should require that many values to accurately define the surface. If that is the case, a more sophisticated program should be considered for the investigation. As mentioned above, the computer generated approximations to airfoil shapes are limited to the NACA XXXX and 230XX series. The program will accept values for ALPHA up to 90 degrees, but the user is cautioned that since separation usually begins at about 10 to 15 degrees, results for values above 15 may be suspect.

## Sample Problem

A few sample problems will illustrate the use of the PANEL program. The first run will be done using an approximation to a NACA 0012 airfoil which is generated by the

program using the information associated with each digit in the NACA number. The second run will analyze a NASA LS(1)-0013 airfoil using a set of data statements containing the airfoil surface coordinates. These statements must be inserted into the proper location in the program prior to running it.

**Starting the Program**

Begin with the screen showing the DCL prompt, which looks like this.

**$**

Next, ensure that the program is in your directory by typing

**DIR [Return]**

and viewing the files for PANEL.EXE and PANEL.OBJ. If only the PANEL.FOR file exists, you must compile the program by typing,

**FOR PANEL [Return]**

The next step is to link the program by entering,

**LINK PANEL [Return]**

The files PANEL.EXE and PANEL.OBJ will now exist and you will be able to run the program.

**Running the Program**

To run the program, type

**PANEL [Return]**

The program will start and the screen should look similar to what is shown in Figure 20.

```
                 PROGRAM PANEL

     SMITH-HESS (DOUGLAS) PANEL METHOD
     FOR A SINGLE-ELEMENT LIFTING AIRFOIL
     IN TWO-DIMENSIONAL INCOMPRESSIBLE FLOW

 DO YOU WISH TO:
      1) USE AIRFOIL SURFACE COORDINATE DATA VALUES.
      2) HAVE COMPUTER GENERATE AN APPROXIMATION
         FOR A NACA XXXX OR 230XX AIRFOIL SECTION.
      3) QUIT THE PROGRAM.
 ENTER 1, 2, OR 3
```

**Figure 20. Initial Screen for Program PANEL**

For the first case we will have the computer generate an approximation for the shape of a NACA 0012 airfoil, consisting of 20 surface panels, using an algorithm contained in subroutine NACA45. The angle of attack of the onset flow will be six degrees. To use the approximation method, enter

**2 [Return]**

Respond to the request for the number of surface data points by entering

**20 [Return]**

Confirm the number of surface data points you desire by entering

**1 [Return]**

Although the program will allow a different number of upper and lower surface data points, it is recommended that you try and keep them equal. An unequal number of nodes yields trailing-edge panels of unequal length, which lowers the accuracy of the approximation to the Kutta condition. Respond to this question by entering

**1 [Return]**

The next question asks for the NACA number of the airfoil you are considering. For this case we will look at the NACA 0012, so enter

**0012 [Return]**

The screen should now look like what is shown in Figure 21.

```
  ENTER NUMBER OF SURFACE DATA POINTS DESIRED
20
  NUMBER OF SURFACE DATA POINTS TO BE GENERATED =          20

  IS THIS VALUE CORRECT? (YES=1, NO=2)
1

  ARE THE NUMBER OF UPPER AND LOWER SURFACE
  DATA POINTS(NODES) EQUAL? (YES=1, NO=2)
1

  INPUT NACA NUMBER, ANY FOUR-DIGIT OR 230XX SERIES
0012



INPUT ALPHA IN DEGREES (ALPHA > 90 TO EXIT LOOP)
```

**Figure 21. Screen Showing Data for Computer Generated Airfoil**

The program is now ready to perform its calculations. The final piece of information required is the angle of attack, ALPHA. By entering values of ALPHA that are less than 90 degrees, you may look at as many different angle of attack cases as you desire. Entering a value for ALPHA that is greater than 90 degrees will cause the program to stop the present airfoil analysis and provide you with a choice of exiting the program or examining another airfoil. For this case, respond to the question by entering

**6 [Return]**

Following entry of the angle of attack, the program begins the solution process. Values scroll up the screen and are simultaneously being written to the data files. When the solution is complete you should see the screen shown in Figure 22.

```
PROGRAM PANEL RESULTS HAVE BEEN WRITTEN TO FILES:

PBODY.DAT   :  BODY SURFACE COORDINATES
PPRES.DAT   :  SURFACE PRESSURE DISTRIBUTION


DO YOU WISH TO:
      1) MAKE ANOTHER RUN OR
      2) END THIS SESSION
ENTER 1 OR 2.
```

**Figure 22. Run Completion Screen**

Say you have finished your analysis of the NACA 0012 at this point and you want to examine another airfoil. Enter a value of ALPHA that is greater than 90 degrees, such as

**99 [Return]**

A new screen will be presented and the program now asks if you want to make another run. Enter

**1 [Return]**

This time the sample problem will examine a NASA LS(1)-0013 whose coordinates have been entered as data statements in the program. You should now be back at the initial screen and it should look like Figure 20. Since you will be using actual airfoil coordinate data values, enter

**1 [Return]**

The screen shown in Figure 23 now presents you with the three choices available for entering the airfoil surface coordinate data values. You will be using the data statements, so enter

**3 [Return]**

```
DO YOU WISH TO ENTER THE SURFACE COORDINATE VALUES:
     1) FROM A DATA FILE.
     2) FROM THE KEYBOARD.
     3) USING DATA STATEMENTS ALREADY ENTERED
        IN THE MAIN PROGRAM. ** NOTE ** THIS REQUIRES
        THAT PROGRAM BE MODIFIED IN ADVANCE BY MOVING
        DATA STATEMENTS TO THE CORRECT LOCATION.
ENTER 1, 2, OR 3.  (FOR PREVIOUS MENU ENTER 4)
```

**Figure 23. Menu for Surface Coordinate Data Entry Method**

The number of data points has been entered via the data statements, therefore you are not asked that question for this case. For the angle of attack, again enter

**6 [Return]**

As you saw in the previous example, values scroll up the screen. These solutions will be appended to the solutions for the NACA 0012 airfoil. The data files are overwritten only when a new session (from the DCL prompt) is started.

The program now asks if you want to make another run. The session is finished, so enter

**2 [Return]**

This completes the sample problems for the PANEL program. The data files created by these sample runs and the listing for the PANEL program are on the following pages.

## SAMPLE PROBLEM OUTPUT DATA FILES

### Sample problem 1: NACA 0012 Airfoil

DATA FILE: PBODY.DAT

BODY SHAPE

| X | Y |
|---|---|
| 1.0000 | 0.0000 |
| 0.9755 | -0.0034 |
| 0.9045 | -0.0129 |
| 0.7939 | -0.0261 |
| 0.6545 | -0.0404 |
| 0.5000 | -0.0526 |
| 0.3455 | -0.0594 |
| 0.2061 | -0.0577 |
| 0.0955 | -0.0460 |
| 0.0245 | -0.0259 |
| 0.0000 | 0.0000 |
| 0.0245 | 0.0259 |
| 0.0955 | 0.0460 |
| 0.2061 | 0.0577 |
| 0.3455 | 0.0594 |
| 0.5000 | 0.0526 |
| 0.6545 | 0.0404 |
| 0.7939 | 0.0261 |
| 0.9045 | 0.0129 |
| 0.9755 | 0.0034 |

**Sample problem 2: NASA LS(1)-0013 Airfoil**

BODY SHAPE

| X | Y |
|---|---|
| 1.0000 | 0.0000 |
| 0.9000 | -0.0116 |
| 0.8000 | -0.0265 |
| 0.7000 | -0.0420 |
| 0.6000 | -0.0546 |
| 0.5000 | -0.0621 |
| 0.4000 | -0.0645 |
| 0.3000 | -0.0632 |
| 0.2000 | -0.0575 |
| 0.1000 | -0.0454 |
| 0.0753 | -0.0407 |
| 0.0500 | -0.0346 |
| 0.0247 | -0.0261 |
| 0.0126 | -0.0194 |
| 0.0000 | 0.0000 |
| 0.0130 | 0.0189 |
| 0.0250 | 0.0258 |
| 0.0499 | 0.0347 |
| 0.0750 | 0.0408 |
| 0.1000 | 0.0454 |
| 0.2000 | 0.0575 |
| 0.3000 | 0.0631 |
| 0.4000 | 0.0643 |
| 0.5000 | 0.0620 |
| 0.6000 | 0.0545 |
| 0.7000 | 0.0418 |
| 0.8000 | 0.0264 |
| 0.9000 | 0.0117 |

**Sample problem 1: NACA 0012 Airfoil**

DATA FILE: PPRESS.DAT

ANGLE OF ATTACK IN DEGREES = 6.000

PRESSURE DISTRIBUTION

| X | CP |
|---|---|
| 0.9878 | 0.2339 |
| 0.9400 | 0.1316 |
| 0.8492 | 0.0728 |
| 0.7242 | 0.0362 |
| 0.5773 | 0.0155 |
| 0.4227 | 0.0180 |
| 0.2758 | 0.0680 |
| 0.1508 | 0.2129 |
| 0.0600 | 0.5547 |
| 0.0122 | 0.9318 |
| 0.0122 | -2.4438 |
| 0.0600 | -1.7390 |
| 0.1508 | -1.1500 |
| 0.2758 | -0.8021 |
| 0.4227 | -0.5537 |
| 0.5773 | -0.3638 |
| 0.7242 | -0.2101 |
| 0.8492 | -0.0717 |
| 0.9400 | 0.0706 |
| 0.9878 | 0.2339 |

CD = 0.00721 CL = 0.72235 CM =-0.18377 CMC4 =-0.00398

**Sample problem 2: NASA LS(1)-0013 Airfoil**

DATA FILE: PPRESS.DAT

ANGLE OF ATTACK IN DEGREES = 6.000

PRESSURE DISTRIBUTION

| X | CP |
|---|---|
| 0.9500 | 0.1566 |
| 0.8500 | 0.0713 |
| 0.7500 | 0.0003 |
| 0.6500 | -0.0572 |
| 0.5500 | -0.0700 |
| 0.4500 | -0.0332 |
| 0.3500 | 0.0239 |
| 0.2500 | 0.1047 |
| 0.1500 | 0.2627 |
| 0.0877 | 0.3930 |
| 0.0627 | 0.4956 |
| 0.0373 | 0.6714 |
| 0.0186 | 0.8801 |
| 0.0063 | 0.7672 |
| 0.0065 | -2.2382 |
| 0.0190 | -2.6638 |
| 0.0375 | -1.9526 |
| 0.0625 | -1.5750 |
| 0.0875 | -1.3623 |
| 0.1500 | -1.0520 |
| 0.2500 | -0.8380 |
| 0.3500 | -0.7090 |
| 0.4500 | -0.6245 |
| 0.5500 | -0.5094 |
| 0.6500 | -0.3375 |
| 0.7500 | -0.1369 |
| 0.8500 | 0.0365 |
| 0.9500 | 0.1566 |

CD = 0.00324    CL = 0.69366    CM =-0.16505    CMC4 = 0.00750

## PROGRAM PANEL LISTING

```
      PROGRAM PANEL
C
C *** MODIFIED FOR USE ON THE MICROVAX/2000 BY J.A. CAMPBELL (JUL 88)
C     FINAL UPDATES MADE 14 SEP 88 - (JAC)
C *******************************************************************
C     PROGRAM PANEL
C
C        SMITH-HESS (DOUGLAS) PANEL METHOD
C        FOR SINGLE-ELEMENT LIFTING AIRFOIL
C        IN TWO-DIMENSIONAL INCOMPRESSIBLE FLOW
C
C *******************************************************************
C
C       SUBROUTINES QUERY AND CLRSCRN ADDED TO ORIGINAL PROGRAM.
C
C       THE FILE OPEN STATEMENTS WERE ADDED IN LIEU OF THE EXEC FILE
C       METHOD USED ON THE IBM MAINFRAME.
C
C *******************************************************************
C
C     ORIGINAL IBM MAINFRAME PROGRAM WAS ADAPTED FROM JACK MORAN'S BOOK
C     'AN INTRODUCTION TO THEORETICAL AND COMPUTATIONAL AERODYNAMICS',
C     WILEY AND SONS, NEW YORK 1984.  THE LISTING IS FOUND ON PAGE 118.
C
C     PROGRAM FLEXIBILITY AND USER INTERFACE WAS REVISED FOR
C     PROFESSOR J.V. HEALEY BY JOHN CAMPBELL.    APRIL 1988.
C
C *******************************************************************
C
C     THIS PROGRAM PROVIDES THE BODY COORDINATES AND THE SURFACE
C     PRESSURE DISTRIBUTION ABOUT A SINGLE ELEMENT LIFTING AIRFOIL
C     IN TWO-DIMENSIONAL FLOW.
C
C     ESTIMATED VALUES FOR LIFT COEFFICIENT AND THE MOMENT COEFFICIENT
C     ABOUT THE LEADING EDGE AND QUARTER CHORD ARE DETERMINED FROM THE
C     PRESSURE COEFFICIENTS OF EACH PANEL.
C
C     YOU MAY PROVIDE ACTUAL AIRFOIL SURFACE COORDINATE DATA VALUES OR
C     HAVE THE COMPUTER GENERATE AN APPROXIMATION FOR THE COORDINATES
C     OF A NACA XXXX OR 230XX AIRFOIL SECTION.
C
C     IF YOU DESIRE TO ENTER THE SURFACE COORDINATE VALUES, SEVERAL
C     OPTIONS ARE AVAILABLE.  YOU MAY ENTER THEM 1) FROM A DATA FILE,
C     2) FROM THE KEYBOARD OR 3) USING DATA STATEMENTS ALREADY ENTERED
C     AT THE END OF THE MAIN PROGRAM LISTING.
C
C     IF INPUTTING YOUR OWN DATA, REMEMBER TO START AT THE TRAILING EDGE
C     (X/C = 1.0), AND WORK TOWARDS THE LEADING EDGE, ENTERING THE LOWER
C     SIDE FIRST, FOLLOWED BY THE UPPER SURFACE.  DO NOT ENTER THE
C     TRAILING EDGE TWICE.  TRY TO ENTER A SUFFICIENT NUMBER OF POINTS
C     NEAR THE NOSE FOR GOOD RESOLUTION.
C
C *** NOTE:  TO SATISFY THE COROLLARY TO THE KUTTA CONDITION, X VALUES
C            FOR POINTS 2 AND N MUST BE THE SAME.  THIS ENSURES THAT THE
C            LAST PANELS, UPPER AND LOWER, ARE OF EQUAL SIZE. **
C
C            CD IS JUST AN INDICATOR OF NUMERICAL ACCURACY OF THIS
C            PROGRAM. VALUE OF CD SHOULD BE NEAR ZERO.
C
C            IF USING DATA STMTS OR AN INPUT FILE, REMEMBER THE NUMBER
C            OF DATA VALUES AS YOU WILL BE ASKED FOR THIS BY THE PROGRAM.
C
C            USE OF THE DATA STATEMENTS REQUIRES THAT PROGRAM BE
C            MODIFIED IN ADVANCE BY MOVING THEM TO THE CORRECT LOCATION.
C
C *******************************************************************
      INTEGER    NANS
      DIMENSION     Z(100),X(100),Y(100)
C
C *** NOTE: IF YOU CHANGE SIZE OF X AND Y, CHANGE N BELOW ALSO!     ***
C
      DATA X, Y /100*0.,100*0./
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
      COMMON /COF/ A(101,111),KUTTA
      COMMON /NUM/ PI,PI2INV
      COMMON /CPD/ CP(100)
C
C *******************************************************************
C   IF USING DATA STMTS FOR X AND Y VALUES, PLACE LINES HERE.
C *** FOLLOWING DATA IS FOR THE NASA LS(1)-0013 AIRFOIL ***
      DATA NUPPER, NLOWER /14,14/
      DATA (X(I),I=1,28)/1.0,.90,.80,.70,.60,.50,.40,.30,.20,.10,
     1 0.07535,0.05,0.0247,0.01255,0.0,0.01301,0.02505,0.04993,0.07498,
     2 0.10,.20,.30,.40,.50,.60,.70,.80,.90/
C *** NOTE: VALUE FOR TRAILING EDGE IS SET TO 0.000 VS ACT THICKNESS *
      DATA (Y(I),I=1,28)/0.00000,-.01165,-.02654,-.04196,-.05459,
     1         -.06209,-.06453,-.06316,-.05755,-.04543,-.04070,-.03462,
     1         -.02612,-.01938,0.0,.01892,.02583,.03465,.04075,.04541,
     2          .05750,.06307,.06432,.06203,.05446,.04183,.02638,.01172/
C *******************************************************************
      PI        = 3.1415926585
C *** MAKE SURE THAT N CORRESPONDS TO THE SIZE OF X AND Y DIMENSION **
      N = 100
C *******************************************************************
C
```

```
C FOLLOWING LINES FOR INPUT/OUTPUT FILES ADDED BY J.A. CAMPBELL (JUL88)
C OPEN FILE FOR BODY SURFACE COORDINATE OUTPUT
      OPEN (UNIT=11,
     2      FILE= 'PBODY.DAT',
     2      ORGANIZATION= 'SEQUENTIAL',
     2      ACCESS= 'SEQUENTIAL',
     2      RECORDTYPE= 'VARIABLE',
     2      FORM= 'FORMATTED',
     2      STATUS= 'UNKNOWN')
C OPEN FILE FOR PRESSURE COEFFICIENT OUTPUT
      OPEN (UNIT=12,
     2      FILE= 'PPRESS.DAT',
     2      ORGANIZATION= 'SEQUENTIAL',
     2      ACCESS= 'SEQUENTIAL',
     2      RECORDTYPE= 'VARIABLE',
     2      FORM= 'FORMATTED',
     2      STATUS= 'UNKNOWN')
C
   60 CALL INDATA(X,Y,N,NLOWER,NUPPER)
      CALL SETUP(X,Y,N,NLOWER,NUPPER)
  100 PRINT 1000
      READ (5,*) ALPHA
      IF (ALPHA .GT. 90.)   GO TO 200
      COSALF  = COS(ALPHA*PI/180.)
      SINALF  = SIN(ALPHA*PI/180.)
      CALL COFISH(SINALF,COSALF,X,Y,N,NLOWER,NUPPER)
      CALL GAUSS(1)
      CALL VELDIS(SINALF,COSALF,X,Y,N,NLOWER,NUPPER,ALPHA)
      CALL FANDM(SINALF,COSALF,X,Y,N,NLOWER,NUPPER)
      GO TO 100
  200 CONTINUE
C   CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THEN PRINT HEADER
      CALL CLRSCRN
      PRINT *
      PRINT *, ' PROGRAM PANEL RESULTS HAVE BEEN WRITTEN TO FILES:'
      PRINT *
      PRINT *, ' PBODY.DAT    :  BODY SURFACE COORDINATES'
      PRINT *, ' PPRES.DAT    :  SURFACE PRESSURE DISTRIBUTION'
      PRINT *
      PRINT *
C        OPTION TO MAKE ANOTHER RUN
      PRINT *
      PRINT *, ' DO YOU WISH TO:     '
      PRINT *, '        1) MAKE ANOTHER RUN OR'
      PRINT *, '        2) END THIS SESSION'
      PRINT *, ' ENTER 1 OR 2.'
      CALL QUERY (NANS)
      IF (NANS .EQ. 1)  GO TO 60
      STOP
 1000 FORMAT(//////,' INPUT ALPHA IN DEGREES (ALPHA > 90 TO EXIT LOOP) ')
      END
C ****************************************************************
      SUBROUTINE INDATA(X,Y,N,NLOWER,NUPPER)
C ****************************************************************
C
C                     SET PARAMETERS OF BODY SHAPE
C                     FLOW SITUATION, AND NODE DISTRIBUTION
C
C                     USER MUST INPUT
C                          NLOWER = NUMBER OF NODES ON LOWER SURFACE
C                          NUPPER = NUMBER OF NODES ON UPPER SURFACE
C                     PLUS DATA ON BODY AND SUBROUTINE BODY
C
      REAL X(N),Y(N)
      INTEGER NUMPTS,I,STATUS
      CHARACTER*20 INFILE
      INTEGER*2 INFILE_SIZE
      LOGICAL  EXIST
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
C  CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THEN PRINT HEADER
    5 CALL CLRSCRN
      PRINT *
      PRINT *
      PRINT *, '                PROGRAM PANEL  '
      PRINT *
      PRINT *, '      SMITH-HESS (DOUGLAS) PANEL METHOD'
      PRINT *, '      FOR A SINGLE-ELEMENT LIFTING AIRFOIL'
      PRINT *, '      IN TWO-DIMENSIONAL INCOMPRESSIBLE FLOW'
      PRINT *
      PRINT *, ' DO YOU WISH TO:     '
      PRINT *, '        1) USE AIRFOIL SURFACE COORDINATE DATA VALUES.'
      PRINT *, '        2) HAVE COMPUTER GENERATE AN APPROXIMATION'
      PRINT *, '           FOR A NACA XXXX OR 230XX AIRFOIL SECTION.'
      PRINT *, '        3) QUIT THE PROGRAM.'
      PRINT *, ' ENTER 1, 2, OR 3'
      READ (5,*)  NFLAG
      GO TO (10,50,999) NFLAG
C **** ROUTINE TO INPUT SHAPE FROM DATA FILE, KEYBOARD OR DATA STMTS **
   10 CALL CLRSCRN
      PRINT *
      PRINT *, ' DO YOU WISH TO ENTER THE SURFACE COORDINATE VALUES: '
      PRINT *, '        1) FROM A DATA FILE.'
      PRINT *, '        2) FROM THE KEYBOARD.'
      PRINT *, '        3) USING DATA STATEMENTS ALREADY ENTERED'
      PRINT *, '           IN THE MAIN PROGRAM. ** NOTE ** THIS REQUIRES'
      PRINT *, '           THAT PROGRAM BE MODIFIED IN ADVANCE BY MOVING'
      PRINT *, '           DATA STATEMENTS TO THE CORRECT LOCATION.'
      PRINT *, ' ENTER 1, 2, OR 3.  (FOR PREVIOUS MENU ENTER 4)'
   12 READ (5,*)  IFLAG
      IF (IFLAG .EQ. 4) GO TO 5
```

```
      IF (IFLAG .LT. 1 .OR. IFLAG .GT. 3)  THEN
         PRINT *, 'INVALID ENTRY. ENTER 1, 2, OR 3.'
         GO TO 12
      END IF
      IF (IFLAG .EQ. 3)  GO TO 100
****  CUE THE USER TO ENTER THE NUMBER OF DATA POINTS (NUMPTS)
   15 CALL CLRSCRN
      PRINT *
      PRINT *, 'ENTER NUMBER OF DATA POINTS'
      READ *, NUMPTS
****  ECHO CHECK THE INPUT
      PRINT *,'NUMBER OF DATA POINTS TO BE ENTERED =',NUMPTS
      PRINT *
      PRINT *,'IS THIS VALUE CORRECT? (YES=1, NO=2)'
      READ *, M1
      IF (M1 .GT. 1) GO TO 15
      CALL NODES(NUMPTS,NLOWER,NUPPER)
C **** SEND CONTROL TO DATA FILE OR KEYBOARD ENTRY ROUTINE ****
      GO TO (20,30,100) IFLAG
*********************************************************************
C *** DATA FILE READ ROUTINE
C
C    LIB$GET_INPUT IS A VAX LIBRARY ROUTINE.  IT MAY BE REPLACED BY AN
C    EQUIVALENT WRITE/READ TO GET THE FILENAME INTO THE PROGRAM.
C
   20 STATUS = LIB$GET_INPUT (INFILE,              ! The input file
     2          'ENTER THE DATA FILE NAME:  ',     ! Prompt
     2                    INFILE_SIZE)             ! Filename size
C  CHECK TO SEE IF THE FILE EXISTS BEFORE TRYING TO ACCESS IT
      IF (INFILE .EQ. '999') GO TO 5
      INQUIRE (FILE = INFILE (1:INFILE_SIZE), EXIST = EXIST)
      IF (.NOT. EXIST) THEN
         PRINT *
         PRINT *, ' THAT FILE NAME DOES NOT EXIST.'
         PRINT *, ' (ENTER 999 TO RETURN TO MENU).'
         PRINT *
         GO TO 20
      END IF
C OPEN FILE FOR SURFACE COORDINATE INPUT
      OPEN (UNIT=13,
     2       FILE= INFILE,
     2       ORGANIZATION= 'SEQUENTIAL',
     2       ACCESS= 'SEQUENTIAL',
     2       RECORDTYPE= 'VARIABLE',
     2       FORM= 'FORMATTED',
     2       STATUS= 'OLD')
      DO 25   I = 1,NUMPTS
         READ (3,*) X(I),Y(I)
      PRINT 1010, X(I),Y(I)
   25 CONTINUE
 1010 FORMAT(F10.4,F10.4)
      GO TO 100
*********************************************************************
C **** ROUTINE TO ENTER DATA FROM THE KEYBOARD *****
   30 CALL INPUT(X,Y,NUMPTS)
      GO TO 100
*********************************************************************
C **** ROUTINE TO CALCULATE SHAPE, GIVEN NACA NUMBER *****
   50 CALL CLRSCRN
      PRINT *
      PRINT *, '  ENTER NUMBER OF SURFACE DATA POINTS DESIRED'
      READ *, NUMPTS
****  ECHO CHECK THE INPUT
      CALL CLRSCRN
      PRINT *
      PRINT *,'  NUMBER OF SURFACE DATA POINTS TO BE GENERATED =',NUMPTS
      PRINT *
      PRINT *,'  IS THIS VALUE CORRECT? (YES=1, NO=2)'
      READ *, M1
      IF (M1 .GT. 1) GO TO 50
      CALL NODES(NUMPTS,NLOWER,NUPPER)
      PRINT *
      PRINT *, '  INPUT NACA NUMBER, ANY FOUR-DIGIT OR 230XX SERIES'
      READ (5,*) NACA
      IEPS     = NACA/1000
      IPTMAX   = NACA/100 - 10*IEPS
      ITAU     = NACA - 1000*IEPS - 100*IPTMAX
      EPSMAX   = IEPS*0.01
      PTMAX    = IPTMAX*0.1
      TAU      = ITAU*0.01
      IF (IEPS .LT. 10) RETURN
      PTMAX    = 0.2025
      EPSMAX   = 2.6595*PTMAX**3
  100 RETURN
  999 STOP
      END
*********************************************************************
      SUBROUTINE NODES(NUMPTS,NLOWER,NUPPER)
*********************************************************************
C
C **** CALCULATE NLOWER AND NUPPER FOR LATER USE ***
      PRINT *
      PRINT *,'  ARE THE NUMBER OF UPPER AND LOWER SURFACE'
      PRINT *,'  DATA POINTS(NODES) EQUAL? (YES=1, NO=2)'
      READ *, M1
      IF (M1 .EQ. 1) THEN
         NLOWER = NUMPTS/2
         NUPPER = NLOWER
      ELSE
         CALL CLRSCRN
         PRINT *
```

```
          PRINT *, ' TOTAL NUMBER OF SURFACE POINTS =', NUMPTS
   20     PRINT *, '-----------------------------------------------'
          PRINT *
          PRINT *, ' INPUT NUMBER OF LOWER SURFACE POINTS, NLOWER'
          READ (5,*)   NLOWER
          PRINT *, ' INPUT NUMBER OF UPPER SURFACE POINTS, NUPPER'
          READ (5,*)   NUPPER
          NTEST = NLOWER + NUPPER
          IF (NTEST .NE. NUMPTS) THEN
           PRINT *, '  OKAY, TRY IT AGAIN EINSTEIN.  REMEMBER ADDITION?'
           PRINT *, '  NLOWER + NUPPER MUST EQUAL',NUMPTS
             GO TO 20
          END IF
      END IF
      RETURN
      END
*****************************************************************
      SUBROUTINE INPUT(A,B,N)
*****************************************************************
      INTEGER N,I
      DIMENSION A(N), B(N)
****  CUE THE USER TO INPUT X VALUES
   10 PRINT *, 'ENTER X VALUES AS MANY PER LINE AS DESIRED'
      READ *, (A(I), I = 1,N)
****  ECHO CHECK THE INPUT
      PRINT 20, N
   20 FORMAT (/1X,'TABLE OF', I3,' X VALUES:'/1X,21('='))
      PRINT 30, (A(I), I=1,N)
   30 FORMAT (1X,3F10.6)
      PRINT *,'ARE THE VALUES CORRECT? (YES=1, NO=2)'
      READ *, J1
      IF (J1 .GT. 1) GO TO 10
****  CUE THE USER TO INPUT Y VALUES
   35 PRINT *,'ENTER Y VALUES AS MANY PER LINE AS DESIRED'
      READ *, (B(J), J=1,N)
****  ECHO CHECK THE INPUT
      PRINT 40, N
   40 FORMAT (/1X,'TABLE OF', I3,' Y VALUES:'/1X,21('='))
      PRINT 30, (B(J), J=1,N)
      PRINT *,'ARE THE VALUES CORRECT? (YES=1, NO=2)'
      READ *, K1
      IF (K1 .GT. 1) GO TO 35
      RETURN
      END
C ***************************************************************
      SUBROUTINE SETUP(X,Y,N,NLOWER,NUPPER)
C ***************************************************************
      REAL X(N),Y(N)
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /NUM/ PI,PI2INV
C     COMMON /SKAL/ NZERO,YMULT
      PI        = 3.1415926585
      PI2INV    = .5/PI
C     NZERO     = 31
C     YMULT     = 200
C
C              SET COORDINATES OF NODES ON BODY SURFACE
C
      PRINT 1000
      WRITE (11,1000)
      NPOINT    = NLOWER
      SIGN      = -1.0
      NSTART    = 0
      DO 110    NSURF = 1,2
      DO 100    N = 1,NPOINT
      FRACT     = FLOAT(N-1)/FLOAT(NPOINT)
      Z         = .5*(1. - COS(PI*FRACT))
      I         = NSTART + N
      IF (NFLAG .EQ. 1) GO TO 90
      CALL BODY(Z,SIGN,X(I),Y(I))
C     CALL PLOTXY(X(I),Y(I))
   90 WRITE (11,1010) X(I),Y(I)
C     PRINT 1010, X(I),Y(I)
  100 CONTINUE
      NPOINT = NUPPER
      SIGN      = 1.0
      NSTART    = NLOWER
  110 CONTINUE
      NODTOT    = NLOWER + NUPPER
      X(NODTOT+1) = X(1)
      Y(NODTOT+1) = Y(1)
C
C              SET SLOPES OF PANELS
C
      DO 200    I = 1,NODTOT
      DX        = X(I+1) - X(I)
      DY        = Y(I+1) - Y(I)
      DIST      = SQRT(DX*DX +DY*DY)
      SINTHE(I)     = DY/DIST
      COSTHE(I)     = DX/DIST
  200 CONTINUE
 1000 FORMAT(/////,' BODY SHAPE',//,4X,'X',9X,'Y',/)
 1010 FORMAT(F10.4,F10.4)
      RETURN
      END
C ***************************************************************
      SUBROUTINE BODY(Z,SIGN,XI,YI)
C ***************************************************************
C
C              RETURN COORDINATES OF POINT ON THE BODY SURFACE
C
```

```
C                  Z = NODE-SPACING PARAMETER
C                  X,Y = CARTESIAN COORDINATES
C                  SIGN = +1. FOR UPPER SURFACE
C                         -1. FOR LOWER SURFACE
C
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
      IF (SIGN .LT. 0.0)       Z = 1. - Z
      CALL NACA45(Z,THICK,CAMBER,BETA)
      XI       = Z - SIGN*THICK*SIN(BETA)
      YI       = CAMBER + SIGN*THICK*COS(BETA)
      RETURN
      END
C **********************************************************************
      SUBROUTINE NACA45(Z,THICK,CAMBER,BETA)
C **********************************************************************
      COMMON /PAR/ NACA,TAU,EPSMAX,PTMAX
C
C              EVALUATE THICKNESS AND CAMBER
C              FOR NACA 4- OR 5-DIGIT AIRFOIL
C
      THICK    = 0.0
      IF (Z .LT. 1.E-10)        GO TO 100
      THICK    = 5.*TAU*(.2969*SQRT(Z) - Z*(.126 + Z*(.3537
     +           - Z*(.2843 - Z*.1015))))
  100 IF (EPSMAX .EQ. 0.0) GO TO 130
      IF (NACA .GT. 9999) GO TO 140
      IF (Z .GT. PTMAX) GO TO 110
      CAMBER   = EPSMAX/PTMAX/PTMAX*(2.*PTMAX - Z)*Z
      DCAMDX   = 2.*EPSMAX/PTMAX/PTMAX*(PTMAX - Z)
      GO TO 120
  110 CAMBER   = EPSMAX/(1.-PTMAX)**2*(1. + Z - 2.*PTMAX)*(1. - Z)
      DCAMDX   = 2.*EPSMAX/(1.-PTMAX)**2*(PTMAX - Z)
  120 BETA     = ATAN(DCAMDX)
      RETURN
  130 CAMBER   = 0.0
      BETA     = 0.0
      RETURN
  140 IF (Z .GT. PTMAX)       GO TO 150
      W        = Z/PTMAX
      CAMBER   = EPSMAX*W*((W - 3.)*W + 3. - PTMAX)
      DCAMDX   = EPSMAX*3.*W*(1. - W)/PTMAX
      GO TO 120
  150 CAMBER   = EPSMAX*(1. - Z)
      DCAMDX   = - EPSMAX
      GO TO 120
      END
C **********************************************************************
      SUBROUTINE COFISH(SINALF,COSALF,X,Y,N,NLOWER,NUPPER)
C **********************************************************************
C
C              SET COEFFICIENTS OF LINEAR SYSTEM
C
      REAL X(N),Y(N)
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /COF/ A(101,111),KUTTA
      COMMON /NUM/ PI,PI2INV
      KUTTA    = NODTOT + 1
C
C              INITIALIZE COEFFICIENTS
C
      DO 90    J = 1,KUTTA
   90 A(KUTTA,J)    = 0.0
C
C              SET VN = 0 AT MID-POINT OF I-TH PANEL
C
      DO 120   I = 1,NODTOT
      XMID     = .5*(X(I) + X(I+1))
      YMID     = .5*(Y(I) + Y(I+1))
      A(I,KUTTA) = 0.0
C
C              --    FIND CONTRIBUTION OF J-TH PANEL
C
      DO 110   J = 1,NODTOT
      FLOG     = 0.0
      FTAN     = PI
      IF (J .EQ. I)        GO TO 100
      DXJ      = XMID - X(J)
      DXJP     = XMID - X(J+1)
      DYJ      = YMID - Y(J)
      DYJP     = YMID - Y(J+1)
      FLOG     = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN     = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
  100 CTIMTJ   = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ   = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      A(I,J)   = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      B        = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
      A(I,KUTTA) = A(I,KUTTA) + B
      IF ((I .GT. 1) .AND. (I .LT. NODTOT))GO TO 110
C
C              --    IF I-TH PANEL TOUCHES TRAILING EDGE,
C                    ADD CONTRIBUTION TO KUTTA CONDITION
C
      A(KUTTA,J) = A(KUTTA,J) - B
      A(KUTTA,KUTTA) = A(KUTTA,KUTTA) + A(I,J)
  110 CONTINUE
C
C              FILL IN KNOWN SIDES
C
      A(I,KUTTA+1) = SINTHE(I)*COSALF - COSTHE(I)*SINALF
  120 CONTINUE
      A(KUTTA,KUTTA+1) = - (COSTHE(1) + COSTHE(NODTOT))*COSALF
```

```
     +                          - (SINTHE(1) + SINTHE(NODTOT))*SINALF
      RETURN
      END
C ******************************************************************
      SUBROUTINE GAUSS(NRHS)
C ******************************************************************
C
C          SOLUTION OF LINEAR ALGEBRAIC SYSTEM BY
C          GAUSS ELIMINATION WITH PARTIAL PIVOTING
C
C               °A          = COEFFICIENT MATRIX
C               NEQNS       = NUMBER OF EQUATIONS
C               NRHS        = NUMBER OF RIGHT HAND SIDES
C
C               RIGHT-HAND SIDES AND SOLUTIONS STORED IN
C               COLUMNS NEQNS+1 THRU NEQNS+NRHS OF °A
C
      COMMON /COF/ A(101,111),NEQNS
      NP       = NEQNS + 1
      NTOT     = NEQNS + NRHS
C
C               GAUSS REDUCTION
C
      DO 150   I = 2,NEQNS
C
C               --  SEARCH FOR LARGEST ENTRY IN (I-1)TH COLUMN
C                   ON OR BELOW MAIN DIAGONAL
C
      IM       = I - 1
      IMAX     = IM
      AMAX     = ABS(A(IM,IM))
      DO 110   J = I,NEQNS
      IF (AMAX .GE. ABS(A(J,IM))) GO TO 110
      IMAX     = J
      AMAX     = ABS(A(J,IM))
  110 CONTINUE
C
C               --   SWITCH (I-1)TH AND IMAXTH EQUATIONS
C
      IF (IMAX .NE. IM)   GO TO 140
      DO 130   J = IM,NTOT
      TEMP     = A(IM,J)
      A(IM,J) = A(IMAX,J)
      A(IMAX,J) = TEMP
  130 CONTINUE
C
C               ELIMINATE (I-1)TH UNKNOWN FROM
C               ITH THRU (NEQNS)TH EQUATIONS
C
  140 DO 150   J = I,NEQNS
      R        = A(J,IM)/A(IM,IM)
      DO 150   K = I,NTOT
  150 A(J,K)   = A(J,K) - R*A(IM,K)
C
C               BACK SUBSTITUTION
C
      DO 220   K = NP,NTOT
      A(NEQNS,K) = A(NEQNS,K)/A(NEQNS,NEQNS)
      DO 210   L = 2,NEQNS
      I        = NEQNS + 1 - L
      IP       = I + 1
      DO 200   J = IP,NEQNS
  200 A(I,K)   = A(I,K) - A(I,J)*A(J,K)
  210 A(I,K)   = A(I,K)/A(I,I)
  220 CONTINUE
      RETURN
      END
C ******************************************************************
      SUBROUTINE VELDIS(SINALF,COSALF,X,Y,N,NLOWER,NUPPER,ALPHA)
C ******************************************************************
C
C               COMPUTE AND PRINT OUT PRESSURE DISTRIBUTION
C
      REAL X(N),Y(N)
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /COF/ A(101,111),KUTTA
      COMMON /CPD/ CP(100)
      COMMON /NUM/ PI,PI2INV
C     COMMON /SKAL/ NZERO,YMULT
      DIMENSION     Q(150)
C     YMULT     = 20.0
      PRINT 1000, ALPHA
      WRITE (12,1000) ALPHA
      PRINT 1005
      WRITE (12,1005)
C
C               RETRIEVE SOLUTION FROM A-MATRIX
C
      DO 50    I = 1,NODTOT
   50 Q(I)     = A(I,KUTTA+1)
      GAMMA    = A(KUTTA,KUTTA+1)
C
C               FIND VTAND CP AT MID-POINT OF I-TH PANEL
C
      DO 130   I = 1,NODTOT
      XMID     = .5*(X(I) + X(I+1))
      YMID     = .5*(Y(I) + Y(I+1))
      VTANG    = COSALF*COSTHE(I) + SINALF*SINTHE(I)
C
C               ADD CONTRIBUTION OF J-TH PANEL
C
```

```
      DO 120  J = 1,NODTOT
      FLOG      = 0.0
      FTAN      = PI
      IF (J .EQ. I) GO TO 100
      DXJ       = XMID - X(J)
      DXJP      = XMID - X(J+1)
      DYJ       = YMID - Y(J)
      DYJP      = YMID - Y(J+1)
      FLOG      = .5*ALOG((DXJP*DXJP+DYJP*DYJP)/(DXJ*DXJ+DYJ*DYJ))
      FTAN      = ATAN2(DYJP*DXJ-DXJP*DYJ,DXJP*DXJ+DYJP*DYJ)
  100 CTIMTJ    = COSTHE(I)*COSTHE(J) + SINTHE(I)*SINTHE(J)
      STIMTJ    = SINTHE(I)*COSTHE(J) - COSTHE(I)*SINTHE(J)
      AA        = PI2INV*(FTAN*CTIMTJ + FLOG*STIMTJ)
      B         = PI2INV*(FLOG*CTIMTJ - FTAN*STIMTJ)
      VTANG     = VTANG - B*Q(J) +GAMMA*AA
  120 CONTINUE
      CP(I)     = 1. - VTANG*VTANG
C     CALL PLOTXY(XMID,CP(I))
      PRINT 1010, XMID,CP(I)
      WRITE (12,1010) XMID,CP(I)
  130 CONTINUE
 1000 FORMAT(/////,' ANGLE OF ATTACK IN DEGREES = ',F8.3,/)
 1005 FORMAT(//,' PRESSURE DISTRIBUTION',//,4X,'X',8X,'CP',/)
 1010 FORMAT(F10.4,F10.4)
      RETURN
      END
C ******************************************************************
      SUBROUTINE FANDM(SINALF,COSALF,X,Y,N,NLOWER,NUPPER)
C ******************************************************************
C
C               COMPUTE AND PRINT OUT CD,CL,CM
C
      REAL X(N),Y(N)
      COMMON /BOD/ NODTOT,COSTHE(100),SINTHE(100),NFLAG
      COMMON /CPD/ CP(100)
      CFX       = 0.0
      CFY       = 0.0
      CM        = 0.0
      CMC4      = 0.0
      DO 100  I = 1,NODTOT
      XMID      = .5*(X(I) + X(I+1))
      YMID      = .5*(Y(I) + Y(I+1))
      DX        = X(I+1) - X(I)
      DY        = Y(I+1) - Y(I)
      CFX       = CFX + CP(I)*DY
      CFY       = CFY - CP(I)*DX
      CM        = CM + CP(I)*(DX*XMID + DY*YMID)
      CMC4      = CMC4 + CP(I)*(DX*(XMID-0.25) + DY*YMID)
  100 CONTINUE
      CD        = CFX*COSALF + CFY*SINALF
      CL        = CFY*COSALF - CFX*SINALF
      PRINT 1000, CD,CL,CM,CMC4
      WRITE (12,1000) CD,CL,CM,CMC4
 1000 FORMAT(/////,'    CD =',F8.5,'    CL =',F8.5,'    CM =',F8.5,
     +'    CMC4 =',F8.5)
      RETURN
      END
C******************************************************************
      SUBROUTINE CLRSCRN
C******************************************************************
C
C  LIBRARY ROUTINE TO CLEAR THE SCREEN.
C
      ISTAT = LIB$ERASE_PAGE (1,1)
      RETURN
      END
C******************************************************************
      SUBROUTINE QUERY(NANS)
C******************************************************************
C
C  ROUTINE TO TRAP ERRORS CAUSED BY IMPROPER RESPONSES TO QUESTIONS.
C  THE COMPUTER GENERATES AND ERROR WHEN A CHARACTER IS SUPPLIED TO
C  A QUESTION EXPECTING AN INTEGER OR REAL VALUE.
C
      NQTEST=0
    1 CONTINUE
      IF (NQTEST .GT. 0) THEN
          PRINT *, '  CHARACTER VALUES ARE NOT VALID.'
          PRINT *, '  PLEASE ENTER A VALUE OF 1 OR 2.'
      END IF
      NQTEST = NQTEST + 1
      READ (5,*,ERR=1)NANS
      RETURN
      END
C ******************************************************************
C     DATA VALUES FOR VARIOUS AIRFOILS. TO USE, REMOVE COMMENTS
C     AND PLACE AFTER COMMON CARDS IN MAIN PROGRAM.
C ******************************************************************
C *** FOLLOWING DATA IS FOR THE NACA 0006 AIRFOIL ***
C     DATA NUPPER, NLOWER /14,14/
C     DATA (X(I),I=1,28)/1.0,.90,.80,.70,.60,.50,.40,.30,.20,.10,
C    1 0.075,0.05,0.025,0.0125,0.0,0.0125,0.025,0.05,0.075,
C    2 0.10,.20,.30,.40,.50,.60,.70,.80,.90/
C *** NOTE: VALUE FOR TRAILING EDGE IS SET TO 0.000 VS ACT THICKNESS *
C     DATA (Y(I),I=1,20)/-.00063,-.00724,-.01312,-.01832,-.02282,
C    1         -.02647,-.02902,-.03001,-.02869,-.02341,0.0,.02341,.02869,
C    2          .03001,.02902,.02647,.02282,.01832,.01312,.00724/
C
C ******************************************************************
C *** FOLLOWING DATA IS FOR THE NACA 0012 AIRFOIL ***
C     DATA NUPPER, NLOWER /14,14/
```

```
C         DATA (X(I),I=1,28)/1.0,.90,.80,.70,.60,.50,.40,.30,.20,.10,
C       1 0.075,0.05,0.025,0.0125,0.0,0.0125,0.025,0.05,0.075,
C       2 0.10,.20,.30,.40,.50,.60,.70,.80,.90/
C   *** NOTE: VALUE FOR TRAILING EDGE IS SET TO 0.000 VS ACT THICKNESS *
C         DATA (Y(I),I=1,28)/0.00000,-.01448,-.02623,-.03664,-.04563,
C       1           -.05294,-.05803,-.06002,-.05737,-.04683,-.04200,-.03555,
C       1           -.02615,-.01894,0.0,.01894,.02615,.03555,.04200,.04683,
C       2            .05737,.06002,.05803,.05294,.04563,.03664,.02623,.01448/
C
C   ****************************************************************
C   *** FOLLOWING DATA IS FOR THE NASA LS(1)-0013 AIRFOIL ***
C         DATA NUPPER, NLOWER /14,14/
C         DATA (X(I),I=1,28)/1.0,.90,.80,.70,.60,.50,.40,.30,.20,.10,
C       1 0.07535,0.05,0.02[illegible]7,0.01255,0.0,0.01301,0.02505,0.04993,0.07498,
C       2 0.10,.20,.30,.40,.50,.60,.70,.80,.90/
C   *** NOTE: VALUE FOR TRAILING EDGE IS SET TO 0.000 VS ACT THICKNESS *
C         DATA (Y(I),I=1,28)/0.00000,-.01165,-.02654,-.04196,-.05459,
C       1           -.06209,-.06453,-.06316,-.05755,-.04543,-.04070,-.03462,
C       1           -.02612,-.01938,0.0,.01892,.02583,.03465,.04075,.04541,
C       2            .05750,.06307,.06432,.06203,.05446,.04183,.02638,.01172/
C   *****************************************************************
*         USER INSTRUCTIONS FOR MANUAL DATA ENTRY:                    *
*                                                                     *
*              (1) UPON CUE ENTER THE TOTAL NUMBER OF AIRFOIL DATA    *
*         POINTS.  DO NOT COUNT THE LEADING OR TRAILING EDGE TWICE.   *
*                                                                     *
*         NOTE: ARRAYS ARE DIMENSIONED TO 100, THIS IS, THEREBY THE   *
*                   LIMITING NUMBER OF DATA POINTS THAT CAN BE ENTERED *
*                   WITHOUT HAVING TO REDIMENSION THE PROGRAMS ARRAYS. *
*                                                                     *
*              (2) ENTER X COORDINATES AS MANY TO A LINE AS DESIRED.  *
*         THE PROGRAM WILL ALLOW FOR CORRECTION IF ANY ERRORS ARE     *
*         MADE. A TABLE OF X COORDINATES IS DISPLAYED FOR THE USER    *
*         TO CHECK HIS INPUT.                                         *
*                                                                     *
*              (3) ENTER Y COORDINATES AS MANY TO A LINE AS DESIRED.  *
*         THE PROGRAM WILL ALLOW FOR CORRECTION IF ANY ERRORS ARE     *
*         MADE. A TABLE OF Y COORDINATES IS DISPLAYED FOR THE USER    *
*         TO CHECK HIS INPUT.                                         *
*                                                                     *
*              (4) PROGRAM ALLOWS FOR AS MANY RUNS AS THE USER DESIRES *
*         SIMPLY FOLLOW CUING SEQUENCE.                               *
*                                                                     *
***********************************************************************
```

# APPENDIX C. PROGRAM VORLAT USER'S MANUAL

## USERS GUIDE CONTENTS

## Introduction

The purpose of the VORLAT program is to provide an application of the vortex lattice method for the determination of the lift distribution of a flat rectangular plate. This method is based on a distribution of discrete horseshoe vortices over a wing surface that has been divided into a finite number of panels. A system of linear equations is developed for the vortex strengths on the panels and solved by matrix methods.

## Assumptions and Limitations

This program is limited to flat rectangular wings. The program divides the wing up into panels using either a uniform grid or cosine spacing method. The cosine spacing algorithm provides a finer grid near the wing tips where the pressure distribution over the wing is rapidly changing. Both methods incorporate an enhancement whereby the panels do not extend to the wing tips, but only to a distance of $\delta/4$ from the tips. The value of $\delta$ is the spanwise width of a wing panel.

The solution is determined for conditions of incompressible and inviscid irrotational flow. Since we are considering an inviscid fluid, the coefficient of drag provided in the results is for the induced drag component only. This program is intended to be used for the analysis of flat rectangular wings with low aspect ratio. High aspect ratio wings are better analyzed using a method based on the lifting line theory.

## Input Description

There are very few input values required for this simple program. Their description and program variable names are listed below.

**AR** - Aspect ratio of the wing. $(\text{Span})^2/\text{Area}$

**NX, NY** - Number of vortices in the X and Y directions.

**ALPHA** - Angle of attack. (Angle between the chord and the freestream velocity.)

**IOPT** - Grid spacing option. Uniform grid or cosine spacing.

## Input Restrictions

The program, as written, is limited to 350 total surface vortices. This may be modified by changing the size of the arrays, however for the wings that this program was intended to analyze, this should be sufficient. The program will accept values for ALPHA up to

45 degrees, but, as noted previously with program PANEL, the user is cautioned that values above 15 may be suspect.

**Sample Problem**

A sample problem will be used to illustrate the use of the VORLAT program. The run will be done using a flat rectangular wing with an aspect ratio of 2. The lattice will be created by placing three vortices on the wing in the X direction and 5 vortices on the wing in the Y direction. The vortices will be distributed using the Uniform Grid spacing option and the wing will be set at an angle of attack (alpha) of 6 degrees.

**Starting the Program**

Begin with the screen showing the DCL prompt, which looks like this.

**$**

Next, ensure that the program is in your directory by typing

**DIR [Return]**

and viewing the files for VORLAT.EXE and VORLAT.OBJ. If only the VORLAT.FOR file exists, you must compile the program by typing,

**FOR VORLAT [Return]**

The next step is to link the program by entering,

**LINK VORLAT [Return]**

The files VORLAT.EXE and VORLAT.OBJ will now exist and you will be able to run the program.

**Running the Program**

To run the program, type

**VORLAT [Return]**

The program will start and the screen should look similar to what is shown in Figure 24

```
PROGRAM VORLAT : VERSION 4 : 10 SEPTEMBER 88

VORTEX-LATTICE METHOD USED TO DETERMINE SPANWISE
LIFT DISTRIBUTION FOR A FLAT RECTANGULAR WING

ENTER THE ASPECT RATIO?
```

**Figure 24. Initial Screen for Program VORLAT**

Respond to the request for the aspect ratio by entering

**2 [Return]**

Respond to the request for the number of vortices by entering

**3,5 [Return]**

Now enter the angle of attack in degrees as

**6 [Return]**

Finally enter the grid spacing option.

**1 [Return]**

The screen is then cleared and you will be presented with what is shown in Figure 25

```
THE CURRENT VALUES ARE:

  1) ASPECT RATIO . . . . . . . . =   2.000000
  2) NUMBER OF VORTICES (NX,NY) =           3         5
  3) ANGLE OF ATTACK (DEGREES)  =   6.000000
  4) GRID SPACING: (1) UNIFORM, (2) COSINE =        1

THE CALCULATED PARAMETERS ARE:

  DELTA X =  0.3333333
  DELTA Y =  0.1904762

  NUMBER OF EQUATIONS TO SOLVE =          15
ARE THESE VALUES CORRECT? (YES=1, NO=2)
```

**Figure 25. Data Review/Correction Screen**

If your display agrees with this, respond to the question by entering

**1 [Return]**

If you should desire to change any values, enter 2, and you will be asked which value you want to correct and the new desired value. Following entry of the correct values and a positive response, the program begins the solution process. It returns with the coefficients of lift and drag at the indicated spanwise positions, as well as the chordwise center of pressure for those positions. Overall values for the coefficients of lift, drag, induced drag and moment about the leading edge are calculated and then printed out near the bottom of the screen. Don't worry if you miss some of the values as they scroll up on the screen. All the values are printed to both the screen and to the data file.

The program now asks if you want to make another run. Enter

**1 [Return]**

You should now be back at the data review/correction screen and it should look like Figure 25. Now run the same wing, but use the cosine grid spacing. Enter

**2 [Return]**

You want to change the grid spacing, so enter

**4 [Return]**

The screen is automatically updated and you will see that the grid spacing has been changed for you also. Since there are only two grid spacings available, the program "knows" to chose the other and this saves you the extra step of having to enter it. Not exactly artificial intelligence, but it helps. You are again asked if the data is correct. As in the previous example, responding with a (1) causes the program to proceed to the output stage. The solution will be printed to the screen and appended to the data file which contains the data from the prior run.

The program now asks if you want to make another run. The session is finished, so enter

**2 [Return]**

This completes the sample problem for the VORLAT program. The data file created by this sample run and the listing for the VORLAT program are on the following pages.

## SAMPLE PROBLEM OUTPUT DATA FILES

```
** UNIFORM GRID SPACING **

 NX=  3  NY=  5    ASPECT RATIO =  2.00    ANGLE OF ATTACK =  6.00

   Y        CL(Y)      CD(Y)     XCP(Y)

0.095    0.32140    0.01232    0.22266
0.286    0.31085    0.01213    0.22061
0.476    0.28791    0.01166    0.21614
0.667    0.24778    0.01068    0.20843
0.857    0.17711    0.00839    0.19624


CL =         0.25620
CD =         0.0105093
CD/CL2 =  0.1601
CMLE =   -0.055004
XCP =        0.21469


** COSINE GRID SPACING **

 NX=  3  NY=  5    ASPECT RATIO =  2.00    ANGLE OF ATTACK =  6.00

   Y        CL(Y)      CD(Y)     XCP(Y)

0.045    0.32155    0.01223    0.22403
0.210    0.31734    0.01220    0.22325
0.476    0.29243    0.01176    0.21844
0.742    0.23258    0.01038    0.20690
0.907    0.14330    0.00733    0.19607


CL =         0.25927
CD =         0.0106156
CD/CL2 =  0.1579
CMLE =   -0.056232
XCP =        0.21688
```

**NOTE: CD/CL2** $= \dfrac{C_{Di}}{C_L^2} = \dfrac{1}{\pi AR}$ Used to compare results to those for elliptic loading.

## PROGRAM VORLAT LISTING

```
      PROGRAM VORLAT
C
C *** MODIFIED FOR USE ON THE MICROVAX/2000 BY J.A. CAMPBELL (JUL 88)
C     FINAL UPDATES MADE 14 SEP 88 - (JAC)
C ******************************************************************
C
C       PROGRAM VORLAT : VERSION 4    4 AUGUST 88
C
C     ORIGINAL IBM MAINFRAME PROGRAM WAS ADAPTED FROM JACK MORAN'S BOOK
C     'AN INTRODUCTION TO THEORETICAL AND COMPUTATIONAL AERODYNAMICS',
C     WILEY AND SONS, NEW YORK 1984.  THE LISTING IS FOUND ON PAGE 151.
C
C     PROGRAM FLEXIBILITY AND USER INTERFACE WAS REVISED FOR
C     PROFESSOR J.V. HEALEY BY JOHN CAMPBELL.  JULY 1988.
C
C     VERSION 4 USES A COSINE SPACING ALGORITHM IN THE SPANWISE
C     DIRECTION OVER A RANGE OF ZERO TO PI OVER THE HALF SPAN
C     AS IMPLEMENTED IN VERSION 3.  AN ATTEMPT WAS MADE TO CHANGE
C     THE RANGE FROM ZERO TO PI/2, HOWEVER TIME CONSTRAINTS PREVENTED
C     ITS SUCCESSFUL INCORPORATION.  THERE APPEARED TO BE A PROBLEM
C     WITH THE INDICES OF THE MATRICES.  THE LINES ADDED IN VERSION 4
C     BUT NOT INCORPORATED ARE NOTED BY CCC IN THE FIRST COLUMNS.
C
C     SIGNIFICANT UPGRADES HAVE BEEN IMPLEMENTED IN VERSION 4 WITH
C     RESPECT TO EASE OF OPERATION AND ERROR CORRECTION.
C
C     STATEMENT LABELS 60, 70, AND 80, AS WELL AS REFERENCES TO THEM,
C     IN THE PREVIOUS VERSION HAVE BEEN CHANGED TO 160, 170 AND 180.
C
C ******************************************************************
C
C       PROGRAM VORLAT : VERSION 3  14 MAY 86
C
C       VORTEX-LATTICE METHOD FOR FLAT RECTANGULAR WING
C
C     **************************************************************
C     ****    VERSION 3 OF THIS PROGRAM INCORPORATES AN OPTION    ****
C     ****    OF HAVING THE COMPUTATION DONE USING EITHER A       ****
C     ****    COSINE GRID SPACING OR UNIFORM GRID SPACING IN      ****
C     ****    BOTH THE X-(CHORDWISE) AND Y-(SPANWISE) DIRECTIONS  ****
C     ****                                                        ****
C     **************************************************************
C
      INTEGER   NANS
      DIMENSION  GAM(350)
      COMMON  DX,DY,AR,PI,IOPT,NX,NY
      COMMON /COF/ A(350,351),NEQNS
      PI    = 3.1415926585
      NPASS = 1
C
C
C FOLLOWING LINES FOR OUTPUT FILES ADDED BY J.A. CAMPBELL (JUL88)
C   OPEN FILE FOR COEFFICIENT OUTPUT
      OPEN (UNIT=11,
     2        FILE= 'VORLAT4.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'UNKNOWN')
C
C           INPUT ASPECT RATIO (AR), NUMBERS OF VORTICES
C           IN X- AND Y- DIRECTIONS (NX,NY) AND
C           ANGLE OF ATTACK IN DEGRESS (ALPHA)
C
C  CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THE PRINT HEADER
      CALL CLRSCRN
      PRINT *
      PRINT *, ' PROGRAM VORLAT : VERSION 4 : 10 SEPTEMBER 88 '
      PRINT *
      PRINT *, ' VORTEX-LATTICE METHOD USED TO DETERMINE SPANWISE'
      PRINT *, ' LIFT DISTRIBUTION FOR A FLAT RECTANGULAR WING'
      PRINT *
      PRINT *
C
   10 PRINT *, ' ENTER THE ASPECT RATIO?'
      READ *, AR
      IF (NPASS .GT. 1)GO TO 70
   30 PRINT *, ' INPUT THE NUMBER OF VORTICES, IN THE X AND Y DIRECTIONS
     + (NX,NY)'
   32 READ *, NX,NY
      IF ((NX*NY) .GT. 350) THEN
         PRINT *, ' NX * NY MUST BE LESS THAN OR EQUAL TO 350.'
         PRINT *, ' PLEASE REENTER.'
         GO TO 32
      END IF
      IF (NPASS .GT. 1)GO TO 70
   50 PRINT *, ' WHAT IS THE ANGLE OF ATTACK IN DEGREES?'
   52 READ *, ALPHA
      IF (ALPHA .EQ. 0.) THEN
         PRINT *,' ALPHA MUST BE GREATER THAN ZERO. PLEASE REENTER.'
         GO TO 52
      ELSE IF (ALPHA .GT. 45.) THEN
         PRINT *,' ALPHA MUST BE LESS THAN 45. PLEASE REENTER.'
         GO TO 52
      END IF
      IF (NPASS .GT. 1)GO TO 72
```

```
   60 PRINT *,' ENTER GRID SPACING OPTION (1 OR 2): (1) UNIFORM',
     +       ', (2) COSINE'
      READ *, IOPT
      NPASS = NPASS + 1
C
C**** MAKE CALCULATIONS AND ECHO CHECK THE INPUT
C
   70 DX = 1./FLOAT(NX)
      DY = AR/(2.*NY + .5)
      NEQNS = NX*NY
C
C  CALL LIBRARY ROUTINE TO CLEAR THE SCREEN
   72 CALL CLRSCRN
C
      PRINT *,' THE CURRENT VALUES ARE: '
      PRINT *
      PRINT *,'     1) ASPECT RATIO . . . . . . . =',AR
      PRINT *,'     2) NUMBER OF VORTICES (NX,NY) =',NX,NY
      PRINT *,'     3) ANGLE OF ATTACK (DEGREES)  =',ALPHA
      PRINT *,'     4) GRID SPACING: (1) UNIFORM, (2) COSINE =',IOPT
      PRINT *
      PRINT *,' THE CALCULATED PARAMETERS ARE: '
      PRINT *
      IF (IOPT .EQ. 1) THEN
         PRINT *,'    DELTA X =',DX
         PRINT *,'    DELTA Y =',DY
      ELSE
         PRINT *,'    SINCE COSINE SPACING WAS CHOSEN,'
         PRINT *,'    DELTA X AND DELTA Y ARE VARIABLE.'
      END IF
      PRINT *
      PRINT *,'     NUMBER OF EQUATIONS TO SOLVE =',NEQNS
      PRINT *
      PRINT *,' ARE THESE VALUES CORRECT? (YES=1, NO=2)'
   75 CALL QUERY (NANS)
      IFLAG = NANS
      IF (IFLAG .LT. 1 .OR. IFLAG .GT. 2)  THEN
          PRINT *, ' INVALID ENTRY. ENTER 1 OR 2.'
          GO TO 75
      END IF
      IF (IFLAG .EQ. 1) GO TO 90
C
      PRINT *, ' WHICH VALUE DO YOU WISH TO CORRECT? '
      PRINT *
   80 PRINT *, ' ENTER 1, 2, 3 OR 4'
      CALL QUERY (NANS)
      IFLAG = NANS
      IF (IFLAG .GT. 4)  THEN
          PRINT *, ' INVALID ENTRY. ENTER 1, 2, 3 OR 4.'
          GO TO 80
      END IF
C **** SEND CONTROL BACK TO OBTAIN CORRECT DATA ****
      GO TO (10,30,50) IFLAG
C **** CHANGE GRID TYPE ****
      IF (IOPT .EQ. 1) THEN
         IOPT = 2
      ELSE
         IOPT = 1
      END IF
      GO TO 72
C
   90 COSALF = COS(ALPHA*PI/180.)
      SINALF = SIN(ALPHA*PI/180.)
C
C    INFORM OPERATOR THAT PROCESSING HAS STARTED
     WRITE (6,1003)
C
C   SET COEFFICIENTS OF EQUATIONS FOR VORTEX STRENGTHS
C
      DO 100 I = 1,NY
        DO 100 J = 1,NX
          IJ = (I - 1)*NX + J
          A(IJ,NEQNS + 1) = SINALF
            DO 100 K = 1,NY
            DO 100 L = 1,NX
              KL = (K-1)*NX + L
              CALL DNWASH (I,J,K,L,A(KL,IJ),1)
  100 CONTINUE
C
C    SOLVE FOR VORTEX STRENGTHS
C
      CALL GAUSS (1)
      DO 200 I = 1,NY
        DO 200 J = 1,NX
          IJ = (I-1)*NX + J
  200 GAM(IJ) = A(IJ,NEQNS+1)
C
C    PRINT OUT HEADINGS FOR DATA
C
      IF (IOPT .EQ. 1) WRITE (11,1000) NX,NY,AR,ALPHA
      IF (IOPT .EQ. 2) WRITE (11,1001) NX,NY,AR,ALPHA
      WRITE (6,1005)
      WRITE (11,1005)
C
C    INITIALIZE TOTAL FORCE AND MOMENT COEFFICIENTS
C
      CMT = 0.0
      CDT = 0.0
      CLT = 0.0
C
```

```
C     COMPUTE FORCE AND MOMENT COEFFICIENTS
C
      DO 320 I = 1,NY
         CX = 0.0
         CZ = 0.0
         CM = 0.0
C
         DO 310 J = 1,NX
            IJ = (I-1)*NX + J
            W = 0.0
              DO 300 K = 1,NY
              DO 300 L = 1,NX
                KL = (K-1)*NX + L
                CALL DNWASH(K,L,I,J,DELW,2)
                W = W + DELW*GAM(KL)
  300         CONTINUE
            CX = CX + GAM(IJ)*(W - SINALF)*2.
            CZ = CZ + GAM(IJ)*COSALF*2.
            IF (IOPT .EQ. 1) THEN
               CM = CM - GAM(IJ)*DX*(J - .75)*COSALF*2.
            ELSE
               CM = CM - GAM(IJ)*(FCOS(J,NX)+0.25*(FCOS(J+1,NX)
     +               - FCOS(J,NX)))*COSALF*2.
            END IF
  310    CONTINUE
         CL = CZ*COSALF - CX*SINALF
         CD = CZ*SINALF + CX*COSALF
         IF (IOPT .EQ. 1) THEN
            CLT =  CLT + CL*DY*2./AR
            CDT = CDT + CD*DY*2./AR
            CMT = CMT + CM*2.*DY/AR
         ELSE
CCC         DELY = (0.5*AR - 0.25*DY)*(FSIN(I+1,NY) - FSIN(I,NY))
            DELY = (0.5*AR - 0.25*DY)*(FCOS(I+1,NY) - FCOS(I,NY))
            CLT = CLT + CL*DELY*2./AR
            CDT = CDT + CD*DELY*2./AR
            CMT = CMT + CM*DELY*2./AR
         END IF
         XCP = - CM/CL
         IF (IOPT .EQ. 1) THEN
            Y = (I-.5)*DY
         ELSE
CCC         Y = (0.5*AR - 0.25*DY)*0.5*(FSIN(I,NY) + FSIN(I+1,NY))
            Y = (0.5*AR - 0.25*DY)*(FCOS(I,NY) +
     +          0.5*(FCOS(I+1,NY) - FCOS(I,NY)))
         END IF
         WRITE(6,1010) Y,CL,CD,XCP
         WRITE(11,1010) Y,CL,CD,XCP
  320 CONTINUE
      XCP = -CMT/CLT
      CDOCL2 = CDT/CLT**2
      WRITE(6,1020) CLT,CDT,CDOCL2,CMT,XCP
      WRITE(11,1020) CLT,CDT,CDOCL2,CMT,XCP
C
      PRINT *
      PRINT *, ' THE COEFFICIENT OUTPUT DATA FOR LIFT, DRAG AND'
      PRINT *, ' PRESSURE HAS BEEN WRITTEN TO FILE VORLAT4.DAT.'
      PRINT *
  400 PRINT *, ' DO YOU WISH TO:     '
      PRINT *, '         1) MAKE ANOTHER RUN OR'
      PRINT *, '         2) END THIS SESSION'
      PRINT *, ' ENTER 1 OR 2.'
      PRINT *
      CALL QUERY (NANS)
      IF (NANS .EQ. 1)  GO TO 70
      STOP
 1000 FORMAT(//,' ** UNIFORM GRID SPACING **',///,
     +       '   NX= ',I2,'  NY= ',I2,'   ASPECT RATIO = ',F5.2,
     &'    ANGLE OF ATTACK = ', F5.2,//)
 1001 FORMAT(//,' ** COSINE GRID SPACING **',///,
     +       '   NX= ',I2,'  NY= ',I2,'   ASPECT RATIO = ',F5.2,
     +'    ANGLE OF ATTACK = ',F5.2,//)
 1003 FORMAT(//,' PROCESSING BEGINS....',///)
 1005 FORMAT (////,'    Y       CL(Y)      CD(Y)      XCP(Y)',/)
 1010 FORMAT(F6.3,3F10.5)
 1020 FORMAT(/////,' CL =',F12.5,/,' CD =',F14.7,/,' CD/CL2 =',F7.4,
     +           /,' CMLE =',F11.6,/,' XCP =',F11.5)
      END
***********************************************************
      SUBROUTINE DNWASH(I,J,K,L,W,IND)
C
C     COMPUTE DOWNWASH ON PANEL CENTERED AT (L-.5)DX,(K-.5)DY
C     DUE TO VORTICES AT PANELS CENTERED AT (J-.5)DX,+-(I-.5)DY
C
      COMMON DX,DY,AR,PI,IOPT,NX,NY
C
      IF (IOPT .EQ. 2) GO TO 50
      XA = DX*(J - .75)
      YA = DY*(I - 1)
      YB = DY*I
      IF (IND .EQ. 1) XP = DX*(L - .25)
      IF (IND .EQ. 2) XP = DX*(L - .75)
      YP = DY*(K-.5)
      GO TO 60
C     THE FOLLOWING LINES HANDLE THE COSINE SPACING SCHEME
C     FAC IS THE HALF SPAN MINUS A 1/4 LATTICE WIDTH INSET.
   50 FAC = 0.5*AR - 0.25*DY
      XA = FCOS(J,NX) + 0.25*(FCOS(J+1,NX) - FCOS(J,NX))
CCC   YA = FAC * FSIN(I-1,NY)
CCC   YB = FAC * FSIN(I,NY)
      YA = FAC * FCOS(I,NY)
```

```
      YB = FAC * FCOS(I+1,NY)
      IF (IND .EQ. 1) XP = FCOS(L,NX) + .75*(FCOS(L+1,NX) - FCOS(L,NX))
      IF (IND .EQ. 2) XP = FCOS(L,NX) + .25*(FCOS(L+1,NX) - FCOS(L,NX))
CCC   YP = FAC*0.5*(FSIN(K,NY) + FSIN(K-1,NY))
      YP = FAC*(FCOS(K,NY) + 0.5*(FCOS(K+1,NY) - FCOS(K,NY)))
C
 60   W = WHV(XP,YP,XA,YA) - WHV(XP,YP,XA, YB)
     +  - WHV(XP,YP,XA,-YA) + WHV(XP,YP,XA,-YB)
      W = W*.25/3.1415926585
      RETURN
      END
*************************************************************************
      FUNCTION WHV(X1,Y1,X2,Y2)
        IF (X1 .EQ. X2)  GO TO 100
        WHV = (1. + SQRT(( X1-X2)**2 + (Y1-Y2)**2)/(X1 - X2))
     +        /(Y1 - Y2)
        RETURN
  100     WHV = 1./(Y1 - Y2)
        RETURN
      END
*************************************************************************
C     THIS RETURNS THE NONDIMENSIONAL X COORD OF EACH SECTION BOUNDARY
C
      FUNCTION FCOS(I,N)
        PI = 3.1415926585
        FRACT = FLOAT(I-1)/FLOAT(N)
        FCOS = 0.5 * (1. - COS(PI*FRACT))
        RETURN
      END
*************************************************************************
C     THIS RETURNS THE NONDIMENSIONAL Y COORD OF EACH SECTION BOUNDARY
C     THIS WAS INTENDED TO IMPLEMENT THE SIN-LAW LATTICE SPACING SCHEME
C     REFERRED TO BY GARY HOUGH, JOU. OF ACFT., MAY 1973, VOL.10, NO.5
C
      FUNCTION FSIN(I,N)
        PI = 3.1415926585
        FRACT = FLOAT(I)/FLOAT(N)
        FSIN =  (SIN(.5*PI*FRACT))
        RETURN
      END
C************************************************************************
      SUBROUTINE CLRSCRN
C
C  LIBRARY ROUTINE TO CLEAR THE SCREEN.
C
      ISTAT = LIB$ERASE_PAGE (1,1)
      RETURN
      END
C************************************************************************
      SUBROUTINE QUERY(NANS)
C
C  ROUTINE TO TRAP ERRORS CAUSED BY IMPROPER RESPONSES TO QUESTIONS.
C  THE COMPUTER GENERATES AND ERROR WHEN A CHARACTER IS SUPPLIED TO
C  A QUESTION EXPECTING AN INTEGER OR REAL VALUE.
C
      NOTEST=0
    1 CONTINUE
      IF (NOTEST .GT. 0) THEN
         PRINT *, '  CHARACTER VALUES ARE NOT VALID.'
         PRINT *, '  PLEASE ENTER AN INTEGER VALUE.'
      END IF
      NOTEST = NOTEST + 1
      READ (5,*,ERR=1)NANS
      RETURN
      END
C************************************************************************
      SUBROUTINE GAUSS (NRHS)
C
C        SOLUTION OF LINEAR ALGEBRAIC SYSTEM BY
C        GAUSS ELIMINATION WITH PARTIAL PIVOTING
C
C               °A          = COEFFICIENT MATRIX
C               NEQNS       = NUMBER OF EQUATIONS
C               NRHS        = NUMBER OF RIGHT HAND SIDES
C
C               RIGHT-HAND SIDES AND SOLUTIONS STORED IN
C               COLUMNS NEQNS+1 THRU NEQNS+NRHS OF °A
C
      COMMON DX,DY,AR,PI
      COMMON /COF/ A(350,351),NEQNS
      NP       = NEQNS + 1
      NTOT     = NEQNS + NRHS
C
C               GAUSS REDUCTION
C
      DO 150  I = 2,NEQNS
C
C               --  SEARCH FOR LARGEST ENTRY IN (I-1)TH COLUMN
C                   ON OR BELOW MAIN DIAGONAL
C
        IM      = I - 1
        IMAX    = IM
        AMAX    = ABS(A(IM,IM))
        DO 110  J = I,NEQNS
          IF (AMAX .GE. ABS(A(J,IM))) GO TO 110
          IMAX    = J
          AMAX    = ABS(A(J,IM))
 110    CONTINUE
C
C               --   SWITCH (I-1)TH AND IMAXTH EQUATIONS
C
```

```
      IF (IMAX .NE. IM)   GO TO 140
      DO 130  J = IM,NTOT
        TEMP    = A(IM,J)
        A(IM,J) = A(IMAX,J)
        A(IMAX,J) = TEMP
130   CONTINUE
C
C           ELIMINATE (I-1)TH UNKNOWN FROM
C           ITH THRU (NEQNS)TH EQUATIONS
C
140   DO 150  J = I,NEQNS
        R = A(J,IM)/A(IM,IM)
      DO 150  K = I,NTOT
150     A(J,K) = A(J,K) - R*A(IM,K)
C
C           BACK SUBSTITUTION
C
      DO 220  K = NP,NTOT
      A(NEQNS,K) = A(NEQNS,K)/A(NEQNS,NEQNS)
      DO 210  L = 2,NEQNS
        I       = NEQNS + 1 - L
        IP      = I + 1
        DO 200  J = IP,NEQNS
200       A(I,K)  = A(I,K) - A(I,J)*A(J,K)
210       A(I,K)  = A(I,K)/A(I,I)
220   CONTINUE
      RETURN
      END
```

# APPENDIX D. PROGRAMS JETFLAP AND JETFLAPIN USER'S MANUAL

## USERS GUIDE CONTENTS

**Introduction**

The purpose of this manual is to permit the user to utilize the JETFLAP program very quickly and easily while requiring little understanding of the underlying EVD theory. The program JETFLAP can be run as a stand alone program if the user wants to develop the JETFLAP input data file manually, but this is not recommended. The layout of the data is not intuitive and its formatting is critical. For this reason, the program JETFLAPIN has been created to assist the user in creating the JETFLAP input data file through an interactive terminal session.

This interactive program is a user-friendly way of creating the input data file required by the wing analysis program JETFLAP. When executed, JETFLAPIN asks questions of the user in order to construct and write to a file the required JETFLAP input data file.

The following manual contains an explanation of the required input data. The reader will find a parallel explanation, with minor modifications, in References 7 and 8. Some parts of these sources have been duplicated in total since they required no comment and were relevant to the present explanation. References to input data cards have been changed to data file lines. In the interest of space, some sections were not included, but the interested reader may find them helpful.

Three sample data input files and their associated output files are included at the end of this appendix. The file VOYTEST.DAT contains information approximating the VOYAGER wing planform. TAPER.DAT illustrates the use of the trapezoidal planform simplification and a semi-circle spacing scheme. The wing is swept 45 degrees, has an aspect ratio of 8.0 and a taper ratio of 0.45. The DOUGLAS.DAT data file is contained in Ref. 7 and was also located at the end of the magnetic tape following the program JETFLAP. It has been used as a program validation test case by comparing the present results with those of Refs. 7 and 8. This file contains information for a simple rectangular jet-flapped wing and three fundamental cases. The stability derivative flag has also been set.

**Assumptions and Limitations**

Before using this program, the user should be aware of the assumptions used in developing the EVD method and the resulting danger of extending the theory beyond its

limits. The assumptions are explained in the section on theory contained in References 7 and 8, but they are summarized below.

1. **Linearity** - This assumption allows the superposition of fundamental geometric cases or solutions but also limits, as an example, the total deflection of flow by flaps. Reference 7 states that the small angle assumptions of the linearized approach make it unlikely that the program would accurately predict the characteristics of a wing with a flap deflected at 60 degrees.
2. **Thin Wing Approximation** - Enabling the simplified treatment of wing sections by transferring boundary conditions to the chordline, this assumption limits the accuracy of the program in modeling thick wings.
3. **Inviscid Flow** - Because of its inability to predict separated flow, the computed lift may be unrealistically large for a wing at high angle of attack or with a sizeable flap deflection. Also, the program cannot consider parasitic drag.
4. **Incompressibility** - This assumption limits the range of speeds for which the program can be used to that in the low subsonic range. The Prandtl-Glauert rule can be applied to cases where subsonic Mach number effects become important (Ref. 3) and, in fact, has been included in a later version of this EVD program.
5. **Irrotationality** - The irrotationality assumption usually imposes no additional limitation in low-speed external aerodynamics where the flow can be considered irrotational.
6. **Interference Effects** - No allowance is made for mutual interference effects between the wing and pylons, nacelles or fuselage. Ground effect is also neglected.
7. **Wing Area Variation** - Although multiple-flapped wings may be modeled, no allowance is made for the increased wing area due to flap extension. An example is a Fowler Flap. If the configuration of concern is such a case, a modification of the original wing planform area input value would have to be made.
8. **Trailing Edge Jet Sheet** - The program only allows the jet sheet to emanate from the wing trailing edge. Therefore, doubtful results will be obtained on augmentor-type flaps, slots and externally blown flap systems.
9. **Computer Run Time** - An increase in the number of elements used to model the wing planform will increase accuracy. However, according to Reference 7 the time to compute increases proportionally between the square and the cube of the number of elements used. On the MicroVAX 2000, a run using 112 elements (VOYAGR.DAT, no jets, two fundamental cases) took 137 seconds to run, while a wing with 37 elements (DOUGLAS.DAT, 21 wing/16 jet elements, three cases and stability derivatives) required only 91 seconds. These times may be further shortened by sending the output to a file vice the screen. In the case of the VOYAGR.DAT run, the time was cut by more than half to a mere 59 seconds.

### Data Preparation Requirements

Prior to using the JETFLAPIN and JETFLAP programs, the user must accomplish the following:

1. Draw a scaled plan view of the wing and, if present, the jets.

2. Divide this planform into spanwise sections parallel to the freestream velocity. A maximum of 40 is permitted.
3. Divide each section into rectangular base elements. These elements, literally, are the bases of the EVD elements [Ref. 8: p. 53] which, in turn, are the "building blocks" of the program operation. Each row can be divided into a maximum of 40 base elements, 20 on the wing and 20 on the jet. However, the maximum number of these elements may not exceed 600.
4. Using a logical scheme, translate the arrangement of these elements and the deflections of the EVD's into a format usable by the program.
5. Refer to the section on the Formulation of the Input Data for a suggested method of approaching the problem of data determination.

### Input Description

A brief description of each piece of input information required is provided during execution of the JETFLAPIN program, however for the benefit of the user they are repeated and expanded upon here.

- **Title Line** - This card provides any desired description of the computer run. The title will be printed at the top of the first page of output. A maximum of 80 characters may be input.
- **General Planform Parameter Line** - This line contains basic planform information.

  **AREA** Wing area, in units of $(SPAN)^2$ to be used for normalization of the aerodynamic coefficients. Must be in the same units as SPAN, i.e., if span is in feet, the area should be in $ft^2$.

  **SPAN** Wing span, in any desired length units.

  **CREF** Wing reference chord, to be used for normalizing various aerodynamic coefficients. It may be any chord length and must be in the same units as SPAN. If a value of 0.0 is input, the mean aerodynamic chord, CMAC, which is computed automatically, will be used.

  **XMC** Pitching moment center. Point about which pitching moments will be taken, measured from the wing apex. Same units as span. NOTE: The wing apex is defined by the program, implicitly, as the intersection of the x-axis with the leading edge when the wing is oriented without a sideslip. If the wing should be input in a yaw, the apex remains at that point.

  **XCG** Wing Center of Gravity. Measured from the apex, this point is used as a pitching axis for computation of stability derivatives, XCG need only be input if IDERIV $\neq$ 0. Same units as SPAN.

- **General Control Line** - This line contains control "flags" which describe the basic characteristics of the computer run.

  **NROWS** Number of spanwise sections (rows) into which the wing is divided. For symmetric or anti-symmetric wings, only the number of sections on the right half of the wing should be input. For non-symmetric wings, NROWS equals the total number of spanwise rows from wing tip to

wing tip. See [Ref. 8: pp. 79-81] for a discussion on symmetric versus non-symmetric wings.

**NCASES** Total Number of Fundamental Cases. There will always be one fundamental case, that being a flat plate at one degree angle of attack. No input data is required for that case and it will be labeled by the program a Case 1. Therefore, NCASES must be one greater than the number of cases for which input data will be given (data lines 12 and 13), to allow for the angle of attack case.

**ISYMM** Symmetry Indicator.

- = 0, Wing is symmetric
- > 0, Wing is non-symmetric
- < 0, Wing is anti-symmetric

**IPRINT** Printed Output Control Flag.

- > 1, Print geometry details and total aerodynamic coefficients,
- = 1, In addition, print spanwise loading,
- = 0, In addition, print chordwise loading,
- < 0, In addition, print all matrices, back substitution checks and other details. This option is normally reserved for trouble-shooting, since it produces a very large amount of output.

**JETFLG** Jet Indicator Flag. A flag used for signaling if there is a jet issuing from the trailing edge of the wing.

- = 0, There is a jet sheet and jet data will be input.
- = 1, There is no jet sheet.

**IGTYPE** Wing Planform Geometry Flag.

- = 1, Wing planform is completely arbitrary and sectional leading and trailing edge coordinates will be read to define the planform.
- = 2, Wing is trapezoidal and simplified planform data will be input. This type of input can only be used if the wing is symmetric. NOTE: Although a triangular shaped wing might be thought of as a degenerative trapezoid, this input cannot be used for a delta planform.

**HINGE** Hinge EVD Flag.

- = 0, Regular EVD's will be used on all hinge elements.
- > 0, Hinge EVD's will be used on all hinge elements. Not permitted if computing dynamic stability derivatives, i.e., IDERIV > 0.

**IDERIV** Dynamic Stability Derivative Flag.

- = 0, Basic run will be executed with no stability derivatives computed.
- > 0, In addition, a dynamic stability derivative run will be executed. This option requires the program to make an additional run, ap-

proximately doubling the computer time. NOTE: The derivative run also reduces to 8 the maximum number of optional fundamental cases permitted, since an extra fundamental case is generated by the program to be used during derivative calculations.

- **Section Centerline Location Lines** - These lines contain the spanwise locations of the centerline of each wing (and jet) section. JETFLAPIN will will place up to eight values on each line, with a maximum of 5 lines (40 sections) allowed.

  **Y** Spanwise distance from wing centerline (x-axis) to the section centerline, normalized by SPAN/2. All values must satisfy ($-1.0 \leq Y \leq 1.0$). NROWS (number of row sections) values must be input, beginning at the right wing tip and working to the wing centerline for symmetric or anti-symmetric wings, or to the left wing tip for non-symmetric wings.

- **Wing Section Type Line** - This card indicates the chordwise arrangement of EVD elements for each section on the wing. The section type is determined by the number and spacing of the elements within each section.

  **ICTYPE** Type Number of Each Wing Section. Any sections having the same number of elements, all with the same distance from the section leading edge (normalized by the xectional chord) are of the same ICTYPE. A maximum of ten different types is allowed. The section at the right wing tip is designated ICTYPE 01. Each new type receives a sequentially higher ICTYPE. The highest ICTYPE is referred to by the program as NWTYPE. NROWS values must be input, therefore, each section must be "typed".

- **Number of Chordwise Wing Elements Line** - This line contains the number of chordwise EVD elements for each wing section type (ICTYPE).

  **NI** Number of Chordwise Elements per ICTYPE. Enter, in ascending order by ICTYPE, the number of elements within that ICTYPE. There may be as few as two or as many as twenty elements per section type. NWTYPE (the number of different section types) values are required.

- **Wing Chordwise Element Coordinates** - These lines contain the x/c coordinates of each EVD element for each ICTYPE.

  **XBW** The chordwise coordinate of each EVD vortex point, measured from the leading edge of the section, normalized by the sectional chord. The first XBW of each set must be 0.0 and the last, less than 1.0. There may be as few as two or as many as twenty values per section type. NWTYPE (the number of different section types) sets of values are required. NOTE: Reference 7, Vol. II refers to these coordinates as XB. The "W" was added in reference 8 to be consistent with the nomenclature of the program listing and also to differentiate between hinge point coord., XBH, and XBJ, the coords. of elements on the jet sheet portion of the section.

- **Planform Information Lines** - There are two types of input lines used to define wing planform. Line 8a is used for arbitrary wing planforms (IGTYPE = 1). Line 8b is used for trapezoidal wing planforms (IGTYPE = 2). The program JETFLAPIN will choose the correct form based on the value of IGTYPE.

- **Leading and Trailing Edge Coordinates** - In order to define an arbitrary planform, the leading and trailing edges for each section must be defined. All section coordinates need not be input, however. The program must have the tip and root coordinates, as a minimum, and any other section's which would define a break in the edge. The program will assume a straight edge exists between coordinates input, and will interpolate between them. A minimum of two sets of coordinates and a maximum of NROWS is required.

  **Y** Spanwise distance from a section centerline to the centerline of the wing, normalized by the half span. Each value must be ***exactly*** the same as those input for the section centerline location lines. JETFLAPIN automatically uses the previously input values.

  **XLEAD** Leading Edge Coordinate. Input the chordwise distance from the section leading edge, **at the section centerline,** to the wing apex. Same units as SPAN, i.e., not normalized by the chord.

  **XTRAIL** Trailing Edge Coordinate. Input the chordwise distance from the section trailing edge, at the section centerline, to the wing apex. Same units as SPAN.

  **9** A "9" must appear in column one of the next line after the last edge coordinate in order to signal that all desired sections have been input. This is required only if IGTYPE = 1 and is handled automatically by JETFLAPIN.

- **Trapezoidal Wing Parameters** - This line contains planform information for the trapezoidal wing. It is used when IGTYPE = 2. This type of input may be used only when the wing planform is symmetric.

  **ARATIO** Wing Aspect Ratio. Input the value of $(SPAN)^2/AREA$. JETFLAPIN automatically calculates this value from previously supplied information.

  **SWEEP** Sweep angle of the Quarter-Chord Line. Input the angle in degrees.

  **TR** Taper Ratio. TR is defined as the chord at the wing tip divided by the chord at the wing root.

- **Jet Section Type Line** - This line indicates the chordwise arrangement of EVD elements for each section on the jet sheet. The jet sheet uses the same sectional boundaries as the conventional wing sections forward of it. This line is required only if JETFLG = 0.

  **IJTYPE** Type Number of Each Jet Section. Input the type number of each section of the wing with respect to the presence of a jet sheet aft of it. Since there is no requirement that the jet sheet span the entire wing, sections without a jet are designated with a "0". Similar to line 5, the wing section type line, as each section within the jet sheet is encountered, it either receives a sequentially higher IJTYPE of the same IJTYPE as a previously labeled equivalent section. The number of different jet section types is NJTYPE. The zeroes do not count as IJTYPE's for the purpose of summing types to find NJTYPE. The number of non-zero values input is NROWSJ, the number of sections having a jet. The maximum number of jet section types is 10. Implied also is that NJTYPE must be less than, or equal to, NROWSJ. NROWS values are required.

NOTE: Due to a computational procedure, there must be at least three adjacent jet sections if there is one. Also, inboard or outboard of a partial span jet sheet, a group of at least three unblown sections must exist.

- **Number of Chordwise Jet Elements** - This line contains the number of chordwise EVD elements for each jet section type. It is similar to line 6, number of chordwise wing elements per section type, except that NJTYPE values must be input. Required only if JETFLG = 0.

  **NI** Number of Chordwise EVD Elements for Each Jet Section Type. Enter, in ascending order by IJTYPE, the number of elements within that IJTYPE. There may be as few as two or as many as ten elements for each jet section type. NJTYPE (the number of different jet section types) values are required.

- **Jet Chordwise Element Coordinates** - These lines contain the x/c coordinates of each element of each jet section type. NJTYPE sets lines are required, each with NI values of x/c. These values are required only if JETFLG = 0.

  **XBJ** Chordwise Coordinate of Each per IJTYPE. The chordwise coordinate of each EVD vortex point, measured from the leading edge of the section (at centerline) and normalized by the sectional chord. Thefirst value for XBW of each set must be 1.0 (trailing edge). The last two base elements in the jet section are overlapped by the Far-Jet (or Jet, or Infinity) EVD which has a length of $10^{10}$, approximating infinity. Therefore, there is no proactical maximum coordinate for elements within the jet. There may be as few as two or as many as ten values per jet section type. NJTYPE (the number of different jet section types) lines of values are required.

- **Fundamental Case Control Line** - This line identifies the types of linear geometric variations to be included in each fundamental case. The number of fundamental cases input must be one less than NCASES (line 3), to allow for the angel of attack case. A separate line is required for each of the input cases. In each of the flags below, a zero value indicates omission of the respective type of input for that fundamental case. A non-zero value indicates that the variation will be included and input must be given to define it. JETFLAPIN sets the non-zero value to correspond with the number of the fundamental case, i.e., for fundamental case number two, variations to be included will be indicated with a "2". For each variation selected, a corresponding line will follow containing the information defining that variation. NOTE: Refs.7 and 8 use the same names shown below, however, in the program listing for JETFLAP under subroutine INCASE they are refered to respectively as INPUTT, INPUTH, INPUTD, INPUTC, and INPUTB.

  **INTWST** Spanswise twist distribution flag.

  **INHITE** Leading edge vertical displacement flag.

  **INDELJ** Jet deflection flag.

  **INCAMB** Camber flag.

  **INBETA** Wing hinge deflection flag.

- **Fundamental Geometric Variation Lines** - These lines are input only in the appropriate flags in the fundamental case control line has been set to a non-zero value.

**TWIST** Sectional Wing Twist. Enter the wing twist, in degrees, at the section centerline, with respect to the wing reference plane. Positive values are in the same sense as a positive angle of attack (leading edge up). NROWS values are required. Required only it INTWST ≠ 0.

**HO** Displacement coordinate of the section leading edge from the wing reference plane, normalized by the sectional chord. Leading edge displacement may be the result of dihedral, twist, nonlinear movement of a leading edge device, etc. Translation resulting from ordinary linear leading and trailing flap deflections and angle of attack are accounted for automatically by the program. These values are used *only* for the computaion of the jet thrust contribution to pitching moments and therefore will have no effect unless jet sheets exist. NROWS values are required. Required only it INHITE ≠ 0.

**DJ** Jet Turning Angle. The jet turning angle, in degrees, relative to the trailing edge. Positive deflection is downward. NOTE: JETFLAPIN requires that the values are input in the order that they are encountered within the jet sheet, working from the right wing tip towards the centerline. NROWSJ values are required. Required only it INDELJ ≠ 0.

**ICT** Camber Type Number for Each Wing Section. These values are similar to the wing section type values on line 5. In order for two sections to have the same ICT, the number of elements, their x/c, and the camber angle associated with them must be the same. NROWS must be input with a maximum of 10 ICT's allowed. The highest value is NCT and there may be no "gaps" in the numbering sequence. A zero value indicates no camber. Required only it INCAMB ≠ 0.

**AC** Camber Angle. The camber angle, in degrees, at eh downwash control point of each EVD. The downwash control point is defined as a point chosen halfway between adjacent XBW's (line 7) including the trailing edge. The angle wi' be positive in the same sense as positive angle of attack. NCT lines are required. Required only it INCAMB ≠ 0.

**ACTE** Trailing Edge Camber Angle. This is the trailing edge deflection angle due to camber only. The values are used for determining the angle at which the jet sheet issues from the wing. These cards are, therefore, only necessary if there is camber (INCAMB ≠ 0) and if there is a jet sheet (JETFLG ≠ 0). NROWSJ values are required.

**IHT** Hinge Section Type. Similar in concept to Wing Type (ICTYPE) and Camber Type (ICT); starting with the first section, designate the type of section with respect to hinges in the section. A section with no hinges will be "0". For sections to have the same IHT they must be alike in their number of hinges, not to exceed four, location of hinges (x/c), their type (leading or trailing edge flap) and in all deflections. There may be as many different IHT's as there are sections. The number of different IHT's is called NHT, and there may be no "gaps" in the sequence.

NROWS values are required, therefore every section must be "typed". Required only it INBETA $\neq$ 0.

**XBH** Hinge Point Distance. The distance from the leading edge of the section to the hinge point, i.e., where the hinge line intersects the section centerline. the distance is normalized by the sectional chord and must be one of the XBW values entered on line 7. A set of values is required for each hinge section type; NHT sets.

**ILT** Leading or Trailing Edge Indicator.

- = 0, Trailing edge flap hinge (positive deflection in the sense of positive angle of attack).
- $\neq$ 0, Leading edge flap hinge (positive deflection in the sense of negative angle of attack).

**BETA** Hinge Deflection Angle. The deflection angle, in degrees, of the element aft of the hinge point relative to the element forward of the hinge point.

- **Composite Case Lines** - These lines indicate how the fundamental cases that are input on lines 12 and 13 are to be combined to form or model the wing under study. A maximum of 24 composite cases are permitted. No composite case may also be chosen and JETFLAPIN will automatically place a "9" in the first column of this line.

N Fundamental Cases to be Included. Indicate the fundamental case number which is to be included in forming a given composite case. As many as ten fundamental cases may be combined in any one composite case. The fundamental cases are identified in the order in which they were input. NOTE: Recall that fundamental case number 1 is the one degree angle of attack case.

A Multiplicative Factor. This factor multiplies the fundamental case previously input. Had the fundamental case included a hinge deflection of 10 degrees, a value of A = 1.6 would introduce a flap deflection of 16 degrees into that particulat composite case.

9 End of Composite Cases. This value is placed at the end of the last composite case or by itself to indicate the completion of composite case information or that no composite cases are desired, respectively. NOTE: This "9" card is not conditional, it will be in every run.

- **Jet Strength Line(s)** - These lines contain the jet strength for all sections which have a jet. An unlimited number of sets of values, maximum of 40 per set, may be entered. NROWSJ values are required. Required only it JETFLG = 0.

**CMU** Sectional Jet Momentum Strength for each jet row. CMU is defined as CMU = J/(qc(y)), where J is the sectional jet momentum per unit span, q is the dynamic pressure, and c(y) is the sectional chord. Since the data refers to only sections with jets, 0.0 may not be input unless all are 0.0. As many sets of CMU data may be input as desired. To run a case on a jet-flapped wing to examine the characteristics without the jet, a set of values all equal to zero must be entered. This option generates a complete set of loadings and other aerodynamic coefficients for each set

of CMU data input. $(0.0 \leq CMU < 800.0)$ Required only if JETFLG = 0.

9 A "9" is placed in column one of the line following all CMU data to signal the end of CMU input. Handled automatically by JETFLAPIN.

### Input Restrictions

A summary of the input restrictions described in References 7 and 8 is listed below. These have been incorporated into the error-checking and screen messages provided in the JETFLAPIN program and are repeated here as a quick reference during data preparation.

1. A "Rule of Three" is implied with regard to dividing the wing (and jet sheet) into sections. At least three adjacent sections of either blown or unblown types are required. A jet cannot consist of one or two sections. Likewise, if the region of jet sheet is partial span and located so that it is bordered on both inboard and outboard sides by conventional (unblown) wing, those unblown portions of the wing must also have three adjacent sections each.
2. The number of spanwise sections, NROWS, requires $3 \leq \text{NROWS} \leq 40$.
3. $1 \leq \text{NCASES} \leq 10$. There is always one Fundamental Case generated by the program. Nine others may be input.
4. The number of chordwise elements in the wing part of a section, NI, requires $2 \leq \text{NI} \leq 20$.
5. The number of chordwise elements in the jet part of a section, NI, requires $2 \leq \text{NI} \leq 20$.
6. Maximum of 10 section types for the wing or the jet. (ICTYPE, IJTYPE $\leq 10$ ).
7. On the wing, $0.0 \leq \text{XBW} \leq 1.0$.
8. On the jet, $1.0 \leq \text{XBJ}$.
9. Only NROWSJ, the number of rows with jets, values required for DJ, ACTE, and CMU.
10. Maximum number of camber section types is 10.
11. There may be as many hinge section types, NHT, as there are rows (or sections). ($1 \leq \text{NHT} \leq \text{NROWS}$)
12. Each section may have four hinges in any combination of leading and trailing edge flaps.
13. The jet blowing coefficient, CMU, is restricted to, $0.0 \leq \text{CMU} \leq 800.0$.

### Formulation of the Input Data

The most difficult and time-consuming part of the wing analysis using the JETFLAP and JETFLAPIN programs is the decomposition of the wing into elements and obtaining the coordinates of those elements. There is hope that follow-on work will be conducted to integrate the sophisticated graphics capabilities of the MicroVAX/2000 with the data input portion of the JETFLAP program, however, for the present, the following methodical approach to the problem is recommended.

A table such as that shown in Ref. 8, p. 117, will help the user organize the required data. Starting at the beginning of the problem, the user is urged to follow the steps below:

1. Make or obtain a scaled drawing of the wing with all flaps and other details drawn on the planform. The scaling is often important in obtaining geometrical data that is often not presented explicitly.
2. If possible, create equations for the leading and trailing edges. For example, if the edge is a straight line, substitute tip and root dimensions into the Two-Point Form of the equation fo a straight line. Such an equation will facilitate the finding of leading trailing edge coordinates once spanwise section centerline coordinates have been established.
3. Draw in spanwise sections taking into account obvious areas of rapidly changing loading (wing tips, near flaps) and rapid changes in sectional chord. It is important to define sections near breaks in the wing, such as leading edge extensions, otherwise the program, seeing only the wing edge coordinates, might read that portion of the leading edge as a relatively flat segment of a multisegment tapered wing.
4. Make two columns, entering sections, starting with 1 at the wing tip, in column one and the section centerline coordinates (normalized by the semi-span) in column two.
5. Draw in chordwise elements for each section. It is more expedient to strive for the same distribution on each section, if possible, unless camber discontinuities (flaps, rapid changes in mean camber ine slope) dictate otherwise.
6. Enter the coordinates of the vortex points, normalized by the *sectional* chord, on each line next ot the appropriate section. **NOTE.** *One of these coordinates must* coincide with the point where the section centerline intersects a flap hinge line, if included. Circle or otherwise mark such coordinates for future identification.
7. Proceeding down the rows of coordinates, any two rows with a different number of values or different values, are of different section types. In ascending order, label in another column each row with its type. The maximum number of types is 10 and the highest type defined is called NWTYPE.
8. At the end of each row write the total number of chordwise elements in that row. Circle the numbers that correspond with different types.
9. In another column, list leading and trailing edge coordinates by substituting (Y) values into the leading and trailing edge equations, if available. **NOTE.** Only those edge coordinates which mark wing root, tip and breaks need be calculated, if the edges are straight line segments.
10. Looking back over the completed table, the data for several of the input data file lines are readily available. Column numbers refer to columns in Table I.

    a. Col. 2 is line 4.

    b. Col. 3 is line 5.

    c. Cols. 4-12 contain data for line(s) 7.

    d. Col. 13 (circled entries) is line 6.

e. Cols. 2, 14, and 15, in that order constitute line 8a.

In addition, the last row number in Col. 1 is NROWS (Cols. 1-2 on line 3). The total of Col. 13 entries is the total number of EVD's, which is limited to 600. More details may be found in Ref. 8 .

**Sample Problem**

A sample session will illustrate the use of the JETFLAP and JETFLAPIN programs. The run can be accomplished using one of the sample data output files provided at the end of this appendix. It is recommended that one of the simpler data files, such as TAPER.DAT or VOYAGR.DAT, be used to respond to the questions asked by the JETFLAPIN program. This method will allow the user to try out the program and get familiar with the questions asked, prior to going through the effort involved in formulating the data for a new problem.

**Starting the Program**

Begin with the screen showing the DCL prompt, which looks like this.

**$**

Next, ensure that the program is in your directory by typing

**DIR [Return]**

and viewing the files for JETFLAP.EXE, JETFLAP.OBJ., JETFLAPIN.EXE and JETFLAPIN.OBJ.

If only the JETFLAP.FOR and JETFLAPIN.FOR files exist, you must compile the programs by typing,

**FOR JETFLAP [Return]** , *and if necessary,*

**FOR JETFLAPIN [Return]**

The next step is to link the programs by entering,

**LINK JETFLAP [Return]** , and again if necessary,

**LINK JETFLAPIN [Return]**

The files JETFLAP.EXE, JETFLAP.OBJ., JETFLAPIN.EXE and JETFLAPIN.OBJ will now exist and you will be able to run the programs.

## Running the Program

To run the program, type

**JETFLAPIN [Return]**

The program will start and the screen will display the header for the interactive program. Using one of the sample data files for the correct values and this appendix to assist you with the terminology, answer each question presented. As you proceed through the JETFLAPIN program, opportunities to review and change input data will be presented. Should it become necessary to change your input data after completing the JETFLAPIN program, you can simply edit the created data file using the VAX EDT editor.

After the JETFLAPIN input program has been run to completion, the file will you created will exist on your directory with the file extension .DAT. This file should be reviewed and compared with the sample file used as a reference. If everything is in order, you should run your data file through the JETFLAP wing analysis program.

The JETFLAP wing analysis program will ask you for the file name of the input data file. It is not necessary to enter the file extension .DAT, but you may do so without any ill effects. The program then asks if you wish to have the output sent to the screen or to a file. If you send the data to a file, the program runs faster and you will have the opportunity to review and print out the data. Sending the data to the screen is a quick way to see if the program is executing properly, but there is no permanent record of the run. At this time, the program is not able to print to both the screen and a file. The program is finished when the DCL ($) prompt returns to the screen.

Several sample input data files, the results of those files after being run through JETFLAP and the listings for the JETFLAP and JETFLAPIN programs are on the following pages.

## JETFLAP INPUT DATA FILE VOYTEST.DAT

```
THIS IS A TEST OF THE INPUT PROGRAM JT77IN USING VOYAGER DATA
59040.0000 1332.0000    0.0000    13.5000    0.0000
16 2 0 0 1 1 0 0
  0.998498  0.989489  0.959459  0.891892  0.792793  0.684685  0.576577  0.468468
  0.400901  0.373874  0.355856  0.346847  0.324324  0.261261  0.162162  0.054054
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 7
  0.000000  0.074100  0.222200  0.370400  0.592600  0.740700  0.888900
  0.998498 13.000000 36.099998
  0.989489 11.900000 36.400002
  0.959459 11.200000 37.200001
  0.891892 10.000000 39.099998
  0.792793  8.300000 41.900002
  0.684685  6.300000 44.900002
  0.576577  4.500000 47.900002
  0.468468  2.300000 51.000000
  0.400901  0.800000 52.599998
  0.373874  0.400000 53.400002
  0.355856  0.100000 53.799999
  0.346847  0.000000 54.000000
  0.324324  0.000000 54.000000
  0.261261  0.000000 54.000000
  0.162162  0.000000 54.000000
  0.054054  0.000000 54.000000
9
 0 0 0 2 0
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
-12.355000 -9.560000 -4.899000  0.764000  5.042000  7.969000  5.412000
9
9
```

## ORIGINAL JETFLAP INPUT DATA FILE VOYAGR.DAT (S. M. WHITE)

```
VOYAGER WING FLAT PLATE AND CAMBERED CASES; 16X7 = 112 ELEMENTS
59040.0   1332.0    0.0       13.5
1601000001010000
.998498   .989489   .959459   .891892   .792793   .684685   .576577   .468468
.400901   .373874   .355856   .346847   .324324   .261261   .162162   .054054
01010101010101010101010101010101
07
0.0       .0741     .2222     .3704     .5926     .7407     .8889
.998498   13.0      36.1
.989489   11.9      36.4
.959459   11.2      37.2
.891892   10.0      39.1
.792793   8.3       41.9
.684685   6.3       44.9
.576577   4.5       47.9
.468468   2.3       51.0
.400901   0.8       52.6
.373874   0.4       53.4
.355856   0.1       53.8
.346847   0.0       54.0
.324324   0.0       54.0
.261261   0.0       54.0
.162162   0.0       54.0
.054054   0.0       54.0
9
00000005
01010101010101010101010101010101
-12.355   -9.560    -4.899    0.764     5.042     7.969     5.412
9
```

# PROGRAM OUTPUT DATA FOR VOYTEST.DAT

```
                    ****************************************
                    *  EVD  JET - WING  COMPUTER  PROGRAM  *
                    ****************************************

          VOYAGER WING FLAT PLATE AND CAMBERED CASES: 16X7 = 112 ELEMENTS

                              USED          INPUT
                   AREA =   0.133106   59040.000000
                   SPAN =   2.000000    1332.000000
                   CREF =   0.069765       0.000000
                    XMC =   0.020270      13.500000
                   CMAC =   0.069765      46.463470
                 ARATIO =  30.051224      30.051224
                    XCG =   0.000000       0.000000

                      NROWS = 16         16
                     NCASES =  1          1
                      ISYMM =  0          0
                     IPRINT =  0          0
                     JETFLG =  1          1
                     IGTYPE =  1          1
                     IHINGE =  0          0

                    NUMBER OF WING ELEMENTS = 112
                    NUMBER OF JET ELEMENTS  =   0
                   TOTAL NUMBER OF ELEMENTS = 112
         ************************************************************************
         *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  1  *
         ************************************************************************
*** SECTION  1 ***  Y =  0.998498  DELTA =  0.001502  XLEAD =  0.019520  XTRAIL =  0.054204  CHORD =  0.034685  TANLE =  0.183333
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.019520   0.022090   0.027226   0.032367   0.040074   0.045210   0.050351
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  2 ***  Y =  0.989489  DELTA =  0.007507  XLEAD =  0.017868  XTRAIL =  0.054655  CHORD =  0.036787  TANLE =  0.183333
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.017868   0.020594   0.026042   0.031494   0.039668   0.045116   0.050568
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  3 ***  Y =  0.959459  DELTA =  0.022523  XLEAD =  0.016817  XTRAIL =  0.055856  CHORD =  0.039039  TANLE =  0.026667
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.016817   0.019710   0.025491   0.031277   0.039951   0.045733   0.051519
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  4 ***  Y =  0.891892  DELTA =  0.045044  XLEAD =  0.015015  XTRAIL =  0.058709  CHORD =  0.043694  TANLE =  0.026212
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.015015   0.018253   0.024724   0.031199   0.040908   0.047379   0.053854
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  5 ***  Y =  0.792793  DELTA =  0.054055  XLEAD =  0.012462  XTRAIL =  0.062913  CHORD =  0.050450  TANLE =  0.025758
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.012462   0.016201   0.023673   0.031149   0.042359   0.049831   0.057308
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  6 ***  Y =  0.684685  DELTA =  0.054053  XLEAD =  0.009459  XTRAIL =  0.067417  CHORD =  0.057958  TANLE =  0.025000
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.009459   0.013754   0.022338   0.030927   0.043805   0.052389   0.060978
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  7 ***  Y =  0.576577  DELTA =  0.054055  XLEAD =  0.006757  XTRAIL =  0.071922  CHORD =  0.065165  TANLE =  0.027778
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.006757   0.011585   0.021236   0.030894   0.045374   0.055025   0.064682
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  8 ***  Y =  0.468468  DELTA =  0.054054  XLEAD =  0.003453  XTRAIL =  0.076577  CHORD =  0.073123  TANLE =  0.031944
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.003453   0.008872   0.019701   0.030538   0.046786   0.057616   0.068452
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
      THIS ROW HAS NO JET
*** SECTION  9 ***  Y =  0.400901  DELTA =  0.013513  XLEAD =  0.001201  XTRAIL =  0.078979  CHORD =  0.077778  TANLE =  0.027778
WING ELEMENTS    NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
            XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
            XI   0.001201   0.006965   0.018483   0.030010   0.047292   0.058811   0.070338
           DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
           EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
          TYPE   20         10         10         10         10         10         10
```

```
       THIS ROW HAS NO JET
*** SECTION 10 ***  Y =  0.373874  DELTA =  0.013514  XLEAD =  0.000601  XTRAIL =  0.080180  CHORD =  0.079580  TANLE =  0.025611
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000601   0.006497   0.018283   0.030077   0.047759   0.059545   0.071339
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 11 ***  Y =  0.355856  DELTA =  0.004504  XLEAD =  0.000150  XTRAIL =  0.080781  CHORD =  0.080631  TANLE =  0.025000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000150   0.006125   0.018066   0.030016   0.047932   0.059873   0.071823
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 12 ***  Y =  0.346847  DELTA =  0.004505  XLEAD =  0.000000  XTRAIL =  0.081081  CHORD =  0.081081  TANLE =  0.000000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000000   0.006008   0.018016   0.030032   0.048049   0.060057   0.072073
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 13 ***  Y =  0.324324  DELTA =  0.018018  XLEAD =  0.000000  XTRAIL =  0.081081  CHORD =  0.081081  TANLE =  0.000000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000000   0.006008   0.018016   0.030032   0.048049   0.060057   0.072073
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 14 ***  Y =  0.261261  DELTA =  0.045045  XLEAD =  0.000000  XTRAIL =  0.081081  CHORD =  0.081081  TANLE =  0.000000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000000   0.006008   0.018016   0.030032   0.048049   0.060057   0.072073
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 15 ***  Y =  0.162162  DELTA =  0.054054  XLEAD =  0.000000  XTRAIL =  0.081081  CHORD =  0.081081  TANLE =  0.000000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000000   0.006008   0.018016   0.030032   0.048049   0.060057   0.072073
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
*** SECTION 16 ***  Y =  0.054054  DELTA =  0.054054  XLEAD =  0.000000  XTRAIL =  0.081081  CHORD =  0.081081  TANLE =  0.000000
WING ELEMENTS     NW =  7      TWIST =  0.000000      HL =  0.000000      THETA S =  0.000000
              XB   0.000000   0.074100   0.222200   0.370400   0.592600   0.740700   0.888900
              XI   0.000000   0.006008   0.018016   0.030032   0.048049   0.060057   0.072073
             DEL   0.074100   0.148100   0.148200   0.222200   0.148100   0.148200   0.111100
             EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
            BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
            TYPE   20         10         10         10         10         10         10
       THIS ROW HAS NO JET
```

```
*************************************
*  SECTIONAL JET BLOWING COEFFICIENTS  *
*************************************
```

| ROW | CMU |
|---|---|
| 1 | 0.000000 |
| 2 | 0.000000 |
| 3 | 0.000000 |
| 4 | 0.000000 |
| 5 | 0.000000 |
| 6 | 0.000000 |
| 7 | 0.000000 |
| 8 | 0.000000 |
| 9 | 0.000000 |
| 10 | 0.000000 |
| 11 | 0.000000 |
| 12 | 0.000000 |
| 13 | 0.000000 |
| 14 | 0.000000 |
| 15 | 0.000000 |
| 16 | 0.000000 |

```
*********************************************
*  CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
*********************************************
```

SECTION 1    Y = 0.998498    CHORD = 0.034685

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.000000 | 0.206017 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.074100 | 0.082606 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.222200 | 0.035549 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.370400 | 0.025035 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 0.592600 | 0.015439 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 6 | 0.740700 | 0.011045 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 7 | 0.888900 | 0.006752 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.296163 |
| 2 | 0.029640 | 0.195265 |
| 3 | 0.044460 | 0.144467 |
| 4 | 0.059280 | 0.109765 |
| 5 | 0.074100 | 0.082606 |

```
*********************************************
*  CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
*********************************************
```

SECTION 2    Y = 0.989489    CHORD = 0.036787

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 0.000000 | 0.257194 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.074100 | 0.150969 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.222200 | 0.074147 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 11 | 0.370400 | 0.049467 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 12 | 0.592600 | 0.029455 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 13 | 0.740700 | 0.020609 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 14 | 0.888900 | 0.011980 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.379302 |
| 2 | 0.029640 | 0.262938 |
| 3 | 0.044460 | 0.209061 |
| 4 | 0.059280 | 0.175306 |
| 5 | 0.074100 | 0.150969 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 3 Y = 0.959459 CHORD = 0.039039

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 0.000000 | 0.320581 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 16 | 0.074100 | 0.195136 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 17 | 0.222200 | 0.102889 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 18 | 0.370400 | 0.072265 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 19 | 0.592600 | 0.044937 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 20 | 0.740700 | 0.031647 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 21 | 0.888900 | 0.018411 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.474177 |
| 2 | 0.029640 | 0.330488 |
| 3 | 0.044460 | 0.264761 |
| 4 | 0.059280 | 0.224079 |
| 5 | 0.074100 | 0.195136 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 4 Y = 0.891892 CHORD = 0.043694

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | 0.000000 | 0.351175 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 23 | 0.074100 | 0.215023 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 24 | 0.222200 | 0.114796 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25 | 0.370400 | 0.081531 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 26 | 0.592600 | 0.051545 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 27 | 0.740700 | 0.036574 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 28 | 0.888900 | 0.021471 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.519682 |
| 2 | 0.029640 | 0.362533 |
| 3 | 0.044460 | 0.290787 |
| 4 | 0.059280 | 0.246475 |
| 5 | 0.074100 | 0.215023 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 5 Y = 0.792793 CHORD = 0.050450

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 29 | 0.000000 | 0.359757 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 30 | 0.074100 | 0.220486 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 31 | 0.222200 | 0.117898 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 32 | 0.370400 | 0.083987 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 33 | 0.592600 | 0.053168 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 34 | 0.740700 | 0.037785 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 35 | 0.888900 | 0.022221 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.532423 |
| 2 | 0.029640 | 0.371476 |
| 3 | 0.044460 | 0.298018 |
| 4 | 0.059280 | 0.252665 |
| 5 | 0.074100 | 0.220486 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 6 Y = 0.684685 CHORD = 0.057958

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 36 | 0.000000 | 0.360147 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 37 | 0.074100 | 0.220753 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 38 | 0.222200 | 0.118077 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 39 | 0.370400 | 0.084037 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 40 | 0.592600 | 0.053260 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 41 | 0.740700 | 0.037862 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 42 | 0.888900 | 0.022260 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.533005 |
| 2 | 0.029640 | 0.371899 |
| 3 | 0.044460 | 0.298358 |
| 4 | 0.059280 | 0.252961 |
| 5 | 0.074100 | 0.220753 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 7 Y = 0.576577 CHORD = 0.065165

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 43 | 0.000000 | 0.359147 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 44 | 0.074100 | 0.220135 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 45 | 0.222200 | 0.117705 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 46 | 0.370400 | 0.083761 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 47 | 0.592600 | 0.053107 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 48 | 0.740700 | 0.037733 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 49 | 0.888900 | 0.022192 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.531526 |
| 2 | 0.029640 | 0.370855 |
| 3 | 0.044460 | 0.297526 |
| 4 | 0.059280 | 0.252255 |
| 5 | 0.074100 | 0.220135 |

************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

************************************************

SECTION 8 Y = 0.468468 CHORD = 0.073125

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | 0.000000 | 0.353019 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 51 | 0.074100 | 0.216279 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 52 | 0.222200 | 0.115580 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 53 | 0.370400 | 0.082200 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 54 | 0.592600 | 0.052032 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 55 | 0.740700 | 0.036978 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 56 | 0.888900 | 0.021711 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.522436 |
| 2 | 0.029640 | 0.364687 |
| 3 | 0.044460 | 0.292390 |
| 4 | 0.059280 | 0.247871 |
| 5 | 0.074100 | 0.216278 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 9   Y = 0.400901   CHORD = 0.077778

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 57 | 0.000000 | 0.343243 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 58 | 0.074100 | 0.210506 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 59 | 0.222200 | 0.113318 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 60 | 0.370400 | 0.080923 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 61 | 0.592600 | 0.051287 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 62 | 0.740700 | 0.036404 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 63 | 0.888900 | 0.021314 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.508011 |
| 2 | 0.029640 | 0.354480 |
| 3 | 0.044460 | 0.284422 |
| 4 | 0.059280 | 0.241180 |
| 5 | 0.074100 | 0.210506 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 10   Y = 0.373874   CHORD = 0.079580

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 64 | 0.000000 | 0.345166 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 65 | 0.074100 | 0.211284 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 66 | 0.222200 | 0.112875 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 67 | 0.370400 | 0.080337 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 68 | 0.592600 | 0.050869 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 69 | 0.740700 | 0.036120 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 70 | 0.888900 | 0.021182 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.510777 |
| 2 | 0.029640 | 0.356305 |
| 3 | 0.044460 | 0.285775 |
| 4 | 0.059280 | 0.242210 |
| 5 | 0.074100 | 0.211284 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 11   Y = 0.355856   CHORD = 0.080631

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 71 | 0.000000 | 0.341790 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 72 | 0.074100 | 0.210832 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 73 | 0.222200 | 0.113276 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 74 | 0.370400 | 0.080521 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 75 | 0.592600 | 0.050893 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 76 | 0.740700 | 0.036075 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 77 | 0.888900 | 0.021087 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.506105 |
| 2 | 0.029640 | 0.353467 |
| 3 | 0.044460 | 0.283949 |
| 4 | 0.059280 | 0.241133 |
| 5 | 0.074100 | 0.210832 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 12   Y = 0.346847   CHORD = 0.081081

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 78 | 0.000000 | 0.343009 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 79 | 0.074100 | 0.211159 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 80 | 0.222200 | 0.113282 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 81 | 0.370400 | 0.080637 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 82 | 0.592600 | 0.050939 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 83 | 0.740700 | 0.036122 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 84 | 0.888900 | 0.021080 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.507824 |
| 2 | 0.029640 | 0.354557 |
| 3 | 0.044460 | 0.284706 |
| 4 | 0.059280 | 0.241653 |
| 5 | 0.074100 | 0.211159 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 13   Y = 0.324324   CHORD = 0.081081

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 85 | 0.000000 | 0.347787 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 86 | 0.074100 | 0.213037 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 87 | 0.222200 | 0.113863 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 88 | 0.370400 | 0.080982 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 89 | 0.592600 | 0.051286 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 90 | 0.740700 | 0.036423 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 91 | 0.888900 | 0.021388 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.514685 |
| 2 | 0.029640 | 0.359071 |
| 3 | 0.044460 | 0.288034 |
| 4 | 0.059280 | 0.244168 |
| 5 | 0.074100 | 0.213037 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 14   Y = 0.261261   CHORD = 0.081081

WING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 92 | 0.000000 | 0.352101 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 93 | 0.074100 | 0.215751 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 94 | 0.222200 | 0.115339 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 95 | 0.370400 | 0.082035 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 96 | 0.592600 | 0.051954 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 97 | 0.740700 | 0.036962 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 98 | 0.888900 | 0.021724 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.521084 |
| 2 | 0.029640 | 0.363553 |
| 3 | 0.044460 | 0.291650 |
| 4 | 0.059280 | 0.247254 |
| 5 | 0.074100 | 0.215751 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 15   Y = 0.162162   CHORD = 0.081081

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 99 | 0.000000 | 0.355045 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 100 | 0.074100 | 0.217624 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 101 | 0.222200 | 0.116396 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 102 | 0.370400 | 0.082842 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 103 | 0.592600 | 0.052495 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 104 | 0.740700 | 0.037337 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 105 | 0.888900 | 0.021960 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.525454 |
| 2 | 0.029640 | 0.366620 |
| 3 | 0.044460 | 0.294130 |
| 4 | 0.059280 | 0.249376 |
| 5 | 0.074100 | 0.217624 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 16 Y = 0.054054 CHORD = 0.081081

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 106 | 0.000000 | 0.356321 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 107 | 0.074100 | 0.218427 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 108 | 0.222200 | 0.116840 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 109 | 0.370400 | 0.083175 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 110 | 0.592600 | 0.052723 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 111 | 0.740700 | 0.037499 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 112 | 0.888900 | 0.022060 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.014820 | 0.527346 |
| 2 | 0.029640 | 0.367946 |
| 3 | 0.044460 | 0.295200 |
| 4 | 0.059280 | 0.250289 |
| 5 | 0.074100 | 0.218427 |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 1 *

| SECTION | Y | LIFT CLG | LIFT CLMU | LIFT CL | INDUCED DRAG CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.998498 | 0.039682 | 0.000000 | 0.039682 | 0.0006926 | 0.0000000 | 0.0005489 | 0.0001437 | 0.0000000 | 0.0006882 | 0.0572168 |
| 2 | 0.989489 | 0.066037 | 0.000000 | 0.066037 | 0.0011526 | 0.0000000 | 0.0008555 | 0.0002971 | 0.0000000 | 0.0012146 | 0.0245104 |
| 3 | 0.959459 | 0.089456 | 0.000000 | 0.089456 | 0.0015613 | 0.0000000 | 0.0013291 | 0.0002322 | 0.0000000 | 0.0017461 | 0.0059317 |
| 4 | 0.891892 | 0.099763 | 0.000000 | 0.099763 | 0.0017412 | 0.0000000 | 0.0015949 | 0.0001463 | 0.0000000 | 0.0021795 | 0.0019683 |
| 5 | 0.792793 | 0.102479 | 0.000000 | 0.102479 | 0.0017686 | 0.0000000 | 0.0016738 | 0.0001148 | 0.0000000 | 0.0025851 | 0.0010643 |
| 6 | 0.684685 | 0.102628 | 0.000000 | 0.102628 | 0.0017912 | 0.0000000 | 0.0016775 | 0.0001137 | 0.0000000 | 0.0029741 | 0.0011086 |
| 7 | 0.576577 | 0.102320 | 0.000000 | 0.102320 | 0.0017858 | 0.0000000 | 0.0016682 | 0.0001177 | 0.0000000 | 0.0033338 | 0.0013367 |
| 8 | 0.468468 | 0.100461 | 0.000000 | 0.100461 | 0.0017534 | 0.0000000 | 0.0016117 | 0.0001416 | 0.0000000 | 0.0036730 | 0.0017259 |
| 9 | 0.400901 | 0.098249 | 0.000000 | 0.098249 | 0.0017148 | 0.0000000 | 0.0015237 | 0.0001911 | 0.0000000 | 0.0038208 | 0.0015511 |
| 10 | 0.373874 | 0.098166 | 0.000000 | 0.098166 | 0.0017133 | 0.0000000 | 0.0015408 | 0.0001725 | 0.0000000 | 0.0039060 | 0.0020781 |
| 11 | 0.355856 | 0.097946 | 0.000000 | 0.097946 | 0.0017095 | 0.0000000 | 0.0015108 | 0.0001986 | 0.0000000 | 0.0039487 | 0.0019957 |
| 12 | 0.346847 | 0.098124 | 0.000000 | 0.098124 | 0.0017126 | 0.0000000 | 0.0015216 | 0.0001910 | 0.0000000 | 0.0039780 | 0.0026141 |
| 13 | 0.324324 | 0.098970 | 0.000000 | 0.098970 | 0.0017274 | 0.0000000 | 0.0015643 | 0.0001631 | 0.0000000 | 0.0040125 | 0.0023514 |
| 14 | 0.261261 | 0.100253 | 0.000000 | 0.100253 | 0.0017497 | 0.0000000 | 0.0016034 | 0.0001464 | 0.0000000 | 0.0040643 | 0.0020319 |
| 15 | 0.162162 | 0.101171 | 0.000000 | 0.101171 | 0.0017658 | 0.0000000 | 0.0016303 | 0.0001355 | 0.0000000 | 0.0041015 | 0.0017824 |
| 16 | 0.054054 | 0.101562 | 0.000000 | 0.101562 | 0.0017726 | 0.0000000 | 0.0016420 | 0.0001306 | 0.0000000 | 0.0041174 | 0.0016699 |
| | TOTAL | 0.100170 | 0.000000 | 0.100170 | 0.0017483 | 0.0000000 | 0.0016073 | 0.0001410 | 0.0000000 | | 0.0000962 |

| SECTION | Y | PITCHING MOMENT CMG | CMMU | CMT | CM | | LIFT CENTER XCP/C | XCL/C | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.998498 | -0.007753 | 0.000000 | 0.000000 | -0.007753 | | 0.195380 | 0.195380 | |
| 2 | 0.989489 | -0.014751 | 0.000000 | 0.000000 | -0.014751 | | 0.223377 | 0.223377 | |
| 3 | 0.959459 | -0.021497 | 0.000000 | 0.000000 | -0.021497 | | 0.240311 | 0.240311 | |
| 4 | 0.891892 | -0.024406 | 0.000000 | 0.000000 | -0.024406 | | 0.244641 | 0.244641 | |
| 5 | 0.792793 | -0.025140 | 0.000000 | 0.000000 | -0.025140 | | 0.245317 | 0.245317 | |
| 6 | 0.684685 | -0.025185 | 0.000000 | 0.000000 | -0.025185 | | 0.245399 | 0.245399 | |
| 7 | 0.576577 | -0.025104 | 0.000000 | 0.000000 | -0.025104 | | 0.245350 | 0.245350 | |
| 8 | 0.468468 | -0.024620 | 0.000000 | 0.000000 | -0.024620 | | 0.245070 | 0.245070 | |
| 9 | 0.400901 | -0.024190 | 0.000000 | 0.000000 | -0.024190 | | 0.246213 | 0.246213 | |
| 10 | 0.373874 | -0.024053 | 0.000000 | 0.000000 | -0.024053 | | 0.245026 | 0.245026 | |
| 11 | 0.355856 | -0.024058 | 0.000000 | 0.000000 | -0.024058 | | 0.245625 | 0.245625 | |
| 12 | 0.346847 | -0.024084 | 0.000000 | 0.000000 | -0.024084 | | 0.245446 | 0.245446 | |
| 13 | 0.324324 | -0.024256 | 0.000000 | 0.000000 | -0.024256 | | 0.245084 | 0.245084 | |
| 14 | 0.261261 | -0.024584 | 0.000000 | 0.000000 | -0.024584 | | 0.245225 | 0.245225 | |
| 15 | 0.162162 | -0.024827 | 0.000000 | 0.000000 | -0.024827 | | 0.245399 | 0.245399 | |
| 16 | 0.054054 | -0.024930 | 0.000000 | 0.000000 | -0.024930 | | 0.245464 | 0.245464 | |
| | TOTAL | -0.031169 | 0.000000 | 0.000000 | -0.031169 | (APEX) | 0.311161 | 0.311161 | (X/CREF) |
| | | -0.002065 | 0.000000 | 0.000000 | -0.002065 | (XMC) | 0.021708 | 0.021708 | (X/B/2) |

* TOTAL AERODYNAMIC COEFFICIENTS *

| | | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | CCLG | 0.1001699 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCLJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCL | 0.1001699 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCDG | 0.0017483 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCDJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCS | 0.0016073 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCD | 0.0001410 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CDITZ | 0.0000962 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMG | -0.0311690 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMT | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCM | -0.0311690 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCP | 0.3111612 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCL | 0.3111612 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCPB | 0.0217081 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCLB | 0.0217081 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMGMC | -0.0020645 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMJMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMTMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCMMC | -0.0020645 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CLLG | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CLLJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CLL | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CNJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CNIMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CCY | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBGR | 0.0435954 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBGL | 0.0435954 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBJR | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBJL | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBR | 0.0435954 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBL | 0.0435954 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CPMBR | 0.4352143 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CPMBL | 0.4352143 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |

```
****************************************************
* THE PROGRAM HAS REACHED NORMAL TERMINATION *
****************************************************
```


```
****************************************************
* THE PROGRAM HAS REACHED NORMAL TERMINATION *
****************************************************
```

## JETFLAP INPUT DATA FILE DOUGLAS.DAT

```
** ONR SAMPLE CASE ***  RECTANGULAR WING   CMU = 1   WITH STABILITY DER
4.500     4.500     1.000     0.250     0.250
4 3 0 0 0 2 1 1
0.9750    0.88750   0.68750   0.2750
1 1 2 1
5 6
0.000     0.100     0.200     0.500     0.900
0.000     0.100     0.200     0.500     0.800     0.900
4.500     0.000     1.000
1 1 1 1
4
1.000     1.100     1.500     3.000
0 0 1 0 0
1.000     1.000     1.000     1.000
0 0 0 0 1
0 0 1 0
0.9000   0 1.000
1 0.00  2 10.00 3 10.00
9
1.000     1.000     1.000     1.000
9
```

# PROGRAM OUTPUT DATA FOR DOUGLAS.DAT

```
                              ****************************************
                              *  EVD  JET - WING  COMPUTER  PROGRAM  *
                              ****************************************

            ** ONR SAMPLE CASE ***  RECTANGULAR WING   CMU = 1   WITH STABILITY DER

                                     USED            INPUT
                         AREA  =     0.888889        4.500000
                         SPAN  =     2.000000        4.500000
                         CREF  =     0.444444        1.000000
                          XMC  =     0.111111        0.250000
                         CMAC  =     0.444444        0.999999
                       ARATIO  =     4.500000        4.500000
                          XCG  =     0.111111        0.250000

                        NROWS  =  4                  4
                       NCASES  =  3                  3
                        ISYMM  =  0                  0
                       IPRINT  =  0                  0
                       JETFLG  =  0                  0
                       IGTYPE  =  2                  2
                       IHINGE  =  0                  1

                     NUMBER OF WING ELEMENTS  =  21
                     NUMBER OF JET ELEMENTS   =  16
                     TOTAL NUMBER OF ELEMENTS =  37
               ************************************************************************
               *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  1  *
               ************************************************************************
*** SECTION  1 ***   Y =  0.975000  DELTA =  0.025000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.900000
             XI   0.000000   0.044444   0.088889   0.222222   0.400000
            DEL   0.100000   0.100000   0.300000   0.400000   0.100000
            EPS   1.000000   1.000000   1.000000   1.000000   1.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  0.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
             XI   0.444444   0.488889   0.666667   1.333333
            DEL   0.100000   0.400000   1.500000**********
           BETA   0.000000   0.000000   0.000000   0.000000
           TYPE   10         10         10         30
*** SECTION  2 ***   Y =  0.887500  DELTA =  0.062500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.900000
             XI   0.000000   0.044444   0.088889   0.222222   0.400000
            DEL   0.100000   0.100000   0.300000   0.400000   0.100000
            EPS   1.000000   1.000000   1.000000   1.000000   1.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  0.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
             XI   0.444444   0.488889   0.666667   1.333333
            DEL   0.100000   0.400000   1.500000**********
           BETA   0.000000   0.000000   0.000000   0.000000
           TYPE   10         10         10         30
*** SECTION  3 ***   Y =  0.687500  DELTA =  0.137500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  6        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.800000   0.900000
             XI   0.000000   0.044444   0.088889   0.222222   0.355556   0.400000
            DEL   0.100000   0.100000   0.300000   0.300000   0.100000   0.100000
            EPS   1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  0.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
             XI   0.444444   0.488889   0.666667   1.333333
            DEL   0.100000   0.400000   1.500000**********
           BETA   0.000000   0.000000   0.000000   0.000000
           TYPE   10         10         10         30
*** SECTION  4 ***   Y =  0.275000  DELTA =  0.275000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.900000
             XI   0.000000   0.044444   0.088889   0.222222   0.400000
            DEL   0.100000   0.100000   0.300000   0.400000   0.100000
            EPS   1.000000   1.000000   1.000000   1.000000   1.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  0.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
             XI   0.444444   0.488889   0.666667   1.333333
            DEL   0.100000   0.400000   1.500000**********
           BETA   0.000000   0.000000   0.000000   0.000000
           TYPE   10         10         10         30
               ************************************************************************
               *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  2  *
               ************************************************************************
*** SECTION  1 ***   Y =  0.975000  DELTA =  0.025000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.900000
            EPS   0.000000   0.000000   0.000000   0.000000   0.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  1.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
           BETA   1.000000   0.000000   0.000000   0.000000
           TYPE   43         10         10         30
*** SECTION  2 ***   Y =  0.887500  DELTA =  0.062500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
             XB   0.000000   0.100000   0.200000   0.500000   0.900000
            EPS   0.000000   0.000000   0.000000   0.000000   0.000000
           BETA   0.000000   0.000000   0.000000   0.000000   0.000000
           TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4    D =  0.888889      DJ =  1.000000      ACTE =  0.000000      THETA =  1.000000
             XB   1.000000   1.100000   1.500000   3.000000
           BETA   1.000000   0.000000   0.000000   0.000000
           TYPE   43         10         10         30
*** SECTION  3 ***   Y =  0.687500  DELTA =  0.137500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  6        TWIST =  0.000000        HL =  0.000000        THETA S =  0.000000
```

```
           XB   0.000000   0.100000   0.200000   0.500000   0.800000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  1.000000     ACTE =  0.000000     THETA =  1.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   1.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  4 ***   Y =  0.275000  DELTA =  0.275000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  1.000000     ACTE =  0.000000     THETA =  1.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   1.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
                  **********************************************************************
                  *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  3  *
                  **********************************************************************
*** SECTION  1 ***   Y =  0.975000  DELTA =  0.025000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  0.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  2 ***   Y =  0.887500  DELTA =  0.062500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  0.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  3 ***   Y =  0.687500  DELTA =  0.137500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  6       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.800000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000   1.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000   1.000000
         TYPE   20         10         10         10         10         42
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  1.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  4 ***   Y =  0.275000  DELTA =  0.275000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS   0.000000   0.000000   0.000000   0.000000   0.000000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  0.000000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
                                  **************************************
                                  *  SECTIONAL JET BLOWING COEFFICIENTS  *
                                  **************************************

                                          ROW      CMU
                                           1     1.000000
                                           2     1.000000
                                           3     1.000000
                                           4     1.000000

                  **********************************************************************
                  *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  4  *
                  **********************************************************************
*** SECTION  1 ***   Y =  0.975000  DELTA =  0.025000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS  -0.400000  -0.200000   0.199999   0.900000   1.400000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  1.500000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  2 ***   Y =  0.887500  DELTA =  0.062500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS  -0.400000  -0.200000   0.199999   0.900000   1.400000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  1.500000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  3 ***   Y =  0.687500  DELTA =  0.137500  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  6       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.800000   0.900000
          EPS  -0.400000  -0.200000   0.199999   0.800000   1.199999   1.400000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10         42
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  1.500000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
*** SECTION  4 ***   Y =  0.275000  DELTA =  0.275000  XLEAD =  0.000000  XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.000000
WING ELEMENTS     NW =  5       TWIST =  0.000000     HL =  0.000000     THETA S =  0.000000
           XB   0.000000   0.100000   0.200000   0.500000   0.900000
          EPS  -0.400000  -0.200000   0.199999   0.900000   1.400000
         BETA   0.000000   0.000000   0.000000   0.000000   0.000000
         TYPE   20         10         10         10         10
 JET ELEMENTS     NJ =  4       D =  0.888889     DJ =  0.000000     ACTE =  0.000000     THETA =  1.500000
           XB   1.000000   1.100000   1.500000   3.000000
         BETA   0.000000   0.000000   0.000000   0.000000
         TYPE   43         10         10         30
                              ************************************************
                              *  CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                              ************************************************
                              SECTION  1       Y =  0.975000      CHORD =  0.444444
```

| | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WING | | | | | | | | | | | | |
| | 1 | 0.000000 | 0.129570 | 0.015231 | 0.006761 | 0.028993 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 2 | 0.100000 | 0.062479 | 0.009627 | 0.004529 | 0.041840 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 3 | 0.200000 | 0.036113 | 0.007270 | 0.003624 | 0.042261 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 4 | 0.500000 | 0.016367 | 0.006730 | 0.004280 | 0.044060 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 5 | 0.900000 | 0.007476 | 0.016049 | 0.005106 | 0.033778 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| JET | | | | | | | | | | | | |
| | 22 | 1.000000 | 0.005187 | 0.059976 | 0.003649 | 0.019742 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 23 | 1.100000 | 0.003331 | 0.009183 | 0.002402 | 0.008359 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 24 | 1.500000 | 0.001203 | 0.001247 | 0.000598 | 0.002077 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 25 | 3.000000 | 0.000202 | 0.000111 | 0.000026 | 0.000261 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 |
|---|---|---|---|---|---|
| 1 | 0.020000 | 0.188371 | 0.022600 | 0.010983 | 0.047722 |
| 2 | 0.040000 | 0.127018 | 0.015844 | 0.007135 | 0.039565 |
| 3 | 0.060000 | 0.097175 | 0.012793 | 0.005832 | 0.038460 |
| 4 | 0.080000 | 0.077455 | 0.010931 | 0.005056 | 0.039619 |
| 5 | 0.100000 | 0.062479 | 0.009627 | 0.004529 | 0.041840 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 2 Y = 0.887500 CHORD = 0.444444

| | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WING | | | | | | | | | | | | |
| | 6 | 0.000000 | 0.185800 | 0.024182 | 0.010750 | 0.056734 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 7 | 0.100000 | 0.107433 | 0.017094 | 0.008019 | 0.068596 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 8 | 0.200000 | 0.064708 | 0.012036 | 0.006499 | 0.067874 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 9 | 0.500000 | 0.029236 | 0.012062 | 0.007998 | 0.070135 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 10 | 0.900000 | 0.012979 | 0.027342 | 0.011413 | 0.051704 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| JET | | | | | | | | | | | | |
| | 26 | 1.000000 | 0.008877 | 0.071213 | 0.008014 | 0.030826 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 27 | 1.100000 | 0.005498 | 0.015626 | 0.004877 | 0.014111 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 28 | 1.500000 | 0.001876 | 0.001990 | 0.000959 | 0.003273 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 29 | 3.000000 | 0.000287 | 0.000155 | 0.000032 | 0.000369 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 |
|---|---|---|---|---|---|
| 1 | 0.020000 | 0.273687 | 0.036243 | 0.016196 | 0.090728 |
| 2 | 0.040000 | 0.189277 | 0.025879 | 0.011673 | 0.072112 |
| 3 | 0.060000 | 0.150051 | 0.021396 | 0.009764 | 0.067293 |
| 4 | 0.080000 | 0.125340 | 0.018803 | 0.008694 | 0.066906 |
| 5 | 0.100000 | 0.107433 | 0.017094 | 0.008019 | 0.068596 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 3 Y = 0.687500 CHORD = 0.444444

| | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WING | | | | | | | | | | | | |
| | 11 | 0.000000 | 0.230760 | 0.034686 | 0.014448 | 0.091838 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 12 | 0.100000 | 0.140897 | 0.025102 | 0.011127 | 0.095908 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 13 | 0.200000 | 0.089584 | 0.019422 | 0.009287 | 0.090417 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 14 | 0.500000 | 0.044098 | 0.018778 | 0.012471 | 0.089255 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 15 | 0.800000 | 0.025147 | 0.027831 | 0.032386 | 0.073930 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 16 | 0.900000 | 0.020025 | 0.035635 | 0.085626 | 0.062236 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| JET | | | | | | | | | | | | |
| | 30 | 1.000000 | 0.013414 | 0.076381 | 0.026293 | 0.039216 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 31 | 1.100000 | 0.008238 | 0.019171 | 0.011354 | 0.019388 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 32 | 1.500000 | 0.002674 | 0.002923 | 0.001175 | 0.004705 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 33 | 3.000000 | 0.000348 | 0.000179 | 0.000040 | 0.000442 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 |
|---|---|---|---|---|---|
| 1 | 0.020000 | 0.341408 | 0.052102 | 0.0216[illegible] | 0.143840 |
| 2 | 0.040000 | 0.238065 | 0.037353 | 0.015327 | 0.110679 |
| 3 | 0.060000 | 0.190841 | 0.031040 | 0.013332 | 0.099851 |
| 4 | 0.080000 | 0.161646 | 0.027436 | 0.011965 | 0.096198 |
| 5 | 0.100000 | 0.140897 | 0.025102 | 0.011127 | 0.095908 |

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 4 Y = 0.275000 CHORD = 0.444444

| | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WING | | | | | | | | | | | | |
| | 17 | 0.000000 | 0.261297 | 0.044339 | 0.011630 | 0.121581 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 18 | 0.100000 | 0.162281 | 0.032006 | 0.007930 | 0.117143 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 19 | 0.200000 | 0.105208 | 0.024659 | 0.005808 | 0.106312 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 20 | 0.500000 | 0.054222 | 0.022370 | 0.004253 | 0.101691 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 21 | 0.900000 | 0.025306 | 0.036868 | 0.002913 | 0.071827 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| JET | | | | | | | | | | | | |
| | 34 | 1.000000 | 0.017243 | 0.078281 | 0.002096 | 0.044615 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 35 | 1.100000 | 0.010686 | 0.020564 | 0.001440 | 0.022860 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 36 | 1.500000 | 0.003524 | 0.003578 | 0.000644 | 0.005990 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 37 | 3.000000 | 0.000403 | 0.000220 | 0.000081 | 0.000522 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 |
|---|---|---|---|---|---|
| 1 | 0.020000 | 0.387135 | 0.066586 | 0.017101 | 0.188460 |
| 2 | 0.040000 | 0.270664 | 0.047716 | 0.012172 | 0.142593 |
| 3 | 0.060000 | 0.217728 | 0.039629 | 0.010023 | 0.126294 |
| 4 | 0.080000 | 0.185226 | 0.035006 | 0.008767 | 0.119492 |
| 5 | [illegible] | 0.162281 | 0.032006 | 0.007930 | 0.117143 |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 1 *

| SECTION | Y | LIFT CLG | LIFT CLMU | LIFT CL | | INDUCED DRAG CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.034284 | 0.017453 | 0.051737 | * | 0.0005984 | 0.0001523 | 0.0002930 | 0.0004577 | 1.0000000 | 0.0082369 | 0.0430782 |
| 2 | 0.887500 | 0.056186 | 0.017453 | 0.073639 | * | 0.0009806 | 0.0001523 | 0.0006025 | 0.0005304 | 1.0000000 | 0.0134616 | 0.0219981 |
| 3 | 0.687500 | 0.076014 | 0.017453 | 0.093467 | * | 0.0013267 | 0.0001523 | 0.0009294 | 0.0005496 | 1.0000000 | 0.0182760 | 0.0120724 |
| 4 | 0.275000 | 0.089566 | 0.017453 | 0.107019 | * | 0.0015632 | 0.0001523 | 0.0011916 | 0.0005259 | 1.0000000 | 0.0216792 | 0.0065520 |
| TOTAL | | 0.078903 | 0.017453 | 0.096356 | * | 0.0013771 | 0.0001523 | 0.0010010 | 0.0005285 | 0.9999999 | | 0.0005049 |

| SECTION | Y | PITCHING MOMENT CMG | CMMU | CMT | CM | | LIFT CENTER XCP/C | XCL/C | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | -0.007640 | -0.017453 | 0.017453 | -0.007640 | * * | 0.222834 | 0.485007 | |
| 2 | 0.887500 | -0.013420 | -0.017453 | 0.017453 | -0.013420 | * * | 0.238857 | 0.419256 | |
| 3 | 0.687500 | -0.019453 | -0.017453 | 0.017453 | -0.019453 | * * | 0.255912 | 0.396057 | |
| 4 | 0.275000 | -0.023831 | -0.017453 | 0.017453 | -0.023831 | * * | 0.266070 | 0.385763 | |
| TOTAL | | -0.020516 | -0.017453 | 0.017453 | -0.020516 | (APEX) | 0.260017 | 0.394053 | (X/CREF) |
| | | -0.000790 | -0.013090 | 0.013090 | -0.000790 | (XMC) | 0.115563 | 0.175135 | (X/B/2) |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 2 *

| SECTION | Y | LIFT CLG | LIFT CLMU | LIFT CL | | INDUCED DRAG CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.013256 | 0.017453 | 0.030710 | * | 0.0000000 | 0.0001523 | 0.0000040 | 0.0001403 | 1.0000000 | 0.0044428 | 0.0255782 |
| 2 | 0.887500 | 0.021353 | 0.017453 | 0.038806 | * | 0.0000000 | 0.0001523 | 0.0000102 | 0.0001421 | 1.0000000 | 0.0069194 | 0.0122945 |
| 3 | 0.687500 | 0.028445 | 0.017453 | 0.045899 | * | 0.0000000 | 0.0001523 | 0.0000210 | 0.0001313 | 1.0000000 | 0.0089616 | 0.0057630 |
| 4 | 0.275000 | 0.033527 | 0.017453 | 0.050980 | * | 0.0000000 | 0.0001523 | 0.0000343 | 0.0001180 | 1.0000000 | 0.0103522 | 0.0022041 |
| TOTAL | | 0.029594 | 0.017453 | 0.047047 | * | 0.0000000 | 0.0001523 | 0.0000261 | 0.0001262 | 0.9999999 | | 0.0001167 |

| SECTION | Y | PITCHING MOMENT CMG | CMMU | CMT | CM | | LIFT CENTER XCP/C | XCL/C |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | -0.007834 | -0.017453 | 0.000000 | -0.025288 | * * | 0.590993 | 0.823444 |

| SECTION | Y | CMG | CMMU | CMT | CM | | XCP/C | XCL/C | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.887500 | -0.012082 | -0.017453 | 0.000000 | -0.029535 | * * | 0.565827 | 0.761099 | |
| 3 | 0.687500 | -0.015161 | -0.017453 | 0.000000 | -0.032615 | * * | 0.533000 | 0.719560 | |
| 4 | 0.275000 | -0.017067 | -0.017453 | 0.000000 | -0.034520 | * * | 0.509043 | 0.677125 | |
| TOTAL | | -0.015458 | -0.017453 | 0.000000 | -0.032911 | (APEX) | 0.522332 | 0.699534 | (X/CREF) |
| | | -0.008059 | -0.013090 | 0.000000 | -0.021149 | (XMC) | 0.232148 | 0.310904 | (X/B/2) |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 3 *

| SECTION | Y | LIFT CLG | LIFT CLMU | LIFT CL | * | INDUCED DRAG CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.004811 | 0.000000 | 0.004811 | * | 0.0000000 | 0.0000000 | 0.0000008 | -0.0000008 | 1.0000000 | 0.0013849 | 0.0050637 |
| 2 | 0.887500 | 0.009230 | 0.000000 | 0.009230 | * | 0.0000000 | 0.0000000 | 0.0000020 | -0.0000020 | 1.0000000 | 0.0026327 | 0.0016948 |
| 3 | 0.687500 | 0.024510 | 0.017453 | 0.041964 | * | 0.0000977 | 0.0001523 | 0.0000036 | 0.0002463 | 1.0000000 | 0.0066422 | 0.0108774 |
| 4 | 0.275000 | 0.005419 | 0.000000 | 0.005419 | * | 0.0000000 | 0.0000000 | 0.0000023 | -0.0000023 | 1.0000000 | 0.0014931 | 0.0095407 |
| TOTAL | | 0.011115 | 0.004800 | 0.015915 | * | 0.0000269 | 0.0000419 | 0.0000025 | 0.0000662 | 0.9999999 | | 0.0000785 |

| SECTION | Y | PITCHING MOMENT CMG | CMMU | CMT | CM | | LIFT CENTER XCP/C | XCL/C | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | -0.002250 | 0.000000 | 0.000000 | -0.002250 | * * | 0.467717 | 0.467717 | |
| 2 | 0.887500 | -0.004613 | 0.000000 | 0.000000 | -0.004613 | * * | 0.499847 | 0.499847 | |
| 3 | 0.687500 | -0.016236 | -0.017453 | 0.001745 | -0.031944 | * * | 0.662426 | 0.802827 | |
| 4 | 0.275000 | -0.001892 | 0.000000 | 0.000000 | -0.001892 | * * | 0.349165 | 0.349165 | |
| TOTAL | | -0.006191 | -0.004800 | 0.000480 | -0.010515 | (APEX) | 0.557334 | 0.690835 | (X/CREF) |
| | | -0.003416 | -0.003600 | 0.000480 | -0.006536 | (XMC) | 0.247704 | 0.307038 | (X/B/2) |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 4 *

| SECTION | Y | LIFT CLG | LIFT CLMU | LIFT CL | * | INDUCED DRAG CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.040381 | 0.026180 | 0.066561 | * | 0.0003055 | 0.0003427 | 0.0000147 | 0.0006335 | 1.0000000 | 0.0102552 | 0.0554085 |
| 2 | 0.887500 | 0.065122 | 0.026180 | 0.091302 | * | 0.0004685 | 0.0003427 | 0.0000562 | 0.0007550 | 1.0000000 | 0.0165145 | 0.0278858 |
| 3 | 0.687500 | 0.086605 | 0.026180 | 0.112785 | * | 0.0005723 | 0.0003427 | 0.0001472 | 0.0007678 | 1.0000000 | 0.0220219 | 0.0145238 |
| 4 | 0.275000 | 0.100914 | 0.026180 | 0.127094 | * | 0.0006315 | 0.0003427 | 0.0002580 | 0.0007162 | 1.0000000 | 0.0257749 | 0.0069873 |
| TOTAL | | 0.089478 | 0.026180 | 0.115658 | * | 0.0005785 | 0.0003427 | 0.0001901 | 0.0007311 | 0.9999999 | | 0.0007208 |

| SECTION | Y | PITCHING MOMENT CMG | CMMU | CMT | CM | | LIFT CENTER XCP/C | XCL/C | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | -0.018667 | -0.026180 | 0.000000 | -0.044847 | * * | 0.462280 | 0.673778 | |
| 2 | 0.887500 | -0.029355 | -0.026180 | 0.000000 | -0.055534 | * * | 0.450760 | 0.608249 | |
| 3 | 0.687500 | -0.037713 | -0.026180 | 0.000000 | -0.063893 | * * | 0.435465 | 0.566507 | |
| 4 | 0.275000 | -0.042616 | -0.026180 | 0.000000 | -0.068796 | * * | 0.422296 | 0.541296 | |
| TOTAL | | -0.038412 | -0.026180 | 0.000000 | -0.064592 | (APEX) | 0.429293 | 0.558475 | (X/CREF) |
| | | -0.016043 | -0.019635 | 0.000000 | -0.035678 | (XMC) | 0.190797 | 0.248211 | (X/B/2) |

LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG. CLQ = 0.089478
PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG. CMQ = -0.038412
PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG. CMCMC = -0.016043

* TOTAL AERODYNAMIC COEFFICIENTS *

| | | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | CCLG | 0.0789025 | 0.0295941 | 0.0111151 | 0.0894784 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCLJ | 0.0174533 | 0.0174533 | 0.0047996 | 0.0261799 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCL | 0.0963558 | 0.0470474 | 0.0159148 | 0.1156583 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCDG | 0.0013771 | 0.0000000 | 0.0000269 | 0.0005785 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCDJ | 0.0001523 | 0.0001523 | 0.0000419 | 0.0003427 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCS | 0.0010010 | 0.0000261 | 0.0000025 | 0.0001901 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCD | 0.0005285 | 0.0001262 | 0.0000662 | 0.0007311 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CDITZ | 0.0005049 | 0.0001167 | 0.0000785 | 0.0007208 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCJ | 0.9999999 | 0.9999999 | 0.9999999 | 0.9999999 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMG | -0.0205160 | -0.0154580 | -0.0061949 | -0.0384124 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMJ | -0.0174533 | -0.0174533 | -0.0047996 | -0.0261799 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMT | 0.0174533 | 0.0000000 | 0.0004800 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCM | -0.0205160 | -0.0329113 | -0.0105145 | -0.0645923 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCP | 0.2600171 | 0.5223324 | 0.5573345 | 0.4292928 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCL | 0.3940530 | 0.6995336 | 0.6908354 | 0.5584753 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCPB | 0.1155531 | 0.2321477 | 0.2477041 | 0.1907967 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CXCLB | 0.1751346 | 0.3109038 | 0.3070379 | 0.2482112 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMGMC | -0.0007904 | -0.0080594 | -0.0034161 | -0.0160428 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMJMC | -0.0130900 | -0.0130900 | -0.0035997 | -0.0196350 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CCMTMC | 0.0130900 | 0.0000000 | 0.0004800 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** | CCMMC | -0.0007904 | -0.0211494 | -0.0065358 | -0.0356778 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CLLG | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CLLJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CLL | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CNJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CNIMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * | CCY | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBGR | 0.0359227 | 0.0134640 | 0.0067121 | 0.0408300 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBGL | 0.0359227 | 0.0134640 | 0.0067121 | 0.0408300 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBJR | 0.0087266 | 0.0087265 | 0.0032998 | 0.0130900 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CSJL | 0.0087266 | 0.0087266 | 0.0032998 | 0.0130900 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CER | 0.0445494 | 0.0221906 | 0.0100119 | 0.0539200 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CBL | 0.0445494 | 0.0221906 | 0.0100119 | 0.0539200 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CPMBR | 0.4623422 | 0.4715644 | 0.6290910 | 0.4662009 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| | CPMBL | 0.4623422 | 0.4716644 | 0.6290910 | 0.4662009 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |

* CHORDWISE LOADING FOR COMPOSITE CASE 1 *

FUNDAMENTAL CASE FACTORS

| A(1) | A(2) | A(3) | A(4) | A(5) | A(6) | A(7) | A(8) | A(9) | A(10) |
|---|---|---|---|---|---|---|---|---|---|
| 0.000000 | 10.000000 | 10.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

*** NOTE *** EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
DO NOT PLOT THESE LOADING POINTS DIRECTLY

SECTION 1 Y = 0.975000 CHORD = 0.444444

| WING | | | | | |
|---|---|---|---|---|---|
| XB | 0.000000 | 0.100000 | 0.200000 | 0.500000 | 0.900000 |
| CP(A=0) | 0.219927 | 0.141560 | 0.108940 | 0.110104 | 0.211551 |
| CP(A=1) | 0.129570 | 0.062479 | 0.036113 | 0.016367 | 0.007476 |

| JET | | | | |
|---|---|---|---|---|
| XB | 1.000000 | 1.100000 | 1.500000 | 3.000000 |
| CP(A=0) | 0.626244 | 0.115851 | 0.018450 | 0.001371 |
| CP(A=1) | 0.005187 | 0.003301 | 0.001203 | 0.000202 |

SECTION 2 Y = 0.887500 CHORD = 0.444444

| WING | | | | | |
|---|---|---|---|---|---|
| XB | 0.000000 | 0.100000 | 0.200000 | 0.500000 | 0.900000 |
| CP(A=0) | 0.349322 | 0.251131 | 0.195344 | 0.200600 | 0.387550 |
| CP(A=1) | 0.185800 | 0.107433 | 0.064708 | 0.029236 | 0.012979 |

| JET | | | | |
|---|---|---|---|---|
| XB | 1.000000 | 1.100000 | 1.500000 | 3.000000 |
| CP(A=0) | 0.792271 | 0.205030 | 0.029491 | 0.001867 |
| CP(A=1) | 0.008877 | 0.005498 | 0.001876 | 0.000287 |

SECTION 3 Y = 0.687500 CHORD = 0.444444

| WING | | | | | | |
|---|---|---|---|---|---|---|
| XB | 0.000000 | 0.100000 | 0.200000 | 0.500000 | 0.800000 | 0.900000 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CP(A=0) | 0.491325 | 0.362287 | 0.287097 | 0.312489 | 0.602165 | 1.212604 |
| CP(A=1) | 0.230760 | 0.140897 | 0.089584 | 0.044098 | 0.025147 | 0.020025 |
| JET | | | | | | |
| XB | 1.000000 | 1.100000 | 1.500000 | 3.000000 | | |
| CP(A=0) | 1.026741 | 0.305251 | 0.042979 | 0.002192 | | |
| CP(A=1) | 0.013414 | 0.008238 | 0.002674 | 0.000248 | | |

SECTION 4 Y = 0.275000 CHORD = 0.444444

| | | | | | |
|---|---|---|---|---|---|
| WING | | | | | |
| XB | 0.000000 | 0.100000 | 0.200000 | 0.500000 | 0.900000 |
| CP(A=0) | 0.557693 | 0.399260 | 0.304669 | 0.266232 | 0.397806 |
| CP(A=1) | 0.261297 | 0.162281 | 0.105208 | 0.054222 | 0.025306 |
| JET | | | | | |
| XB | 1.000000 | 1.100000 | 1.500000 | 3.000000 | |
| CP(A=0) | 0.803748 | 0.220035 | 0.042223 | 0.003008 | |
| CP(A=1) | 0.017243 | 0.010686 | 0.003524 | 0.000403 | |

* COMPOSITE CASE 1 *

FUNDAMENTAL CASE FACTORS

| A(1) | A(2) | A(3) | A(4) | A(5) | A(6) | A(7) | A(8) | A(9) | A(10) |
|---|---|---|---|---|---|---|---|---|---|
| 0.000000 | 10.000000 | 10.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

| SECTION | Y | LIFT CLGO / CLGA | CLMUO / CLMUA | CLO / CLA | PITCHING MOMENT CMGO / CMGA | CMMUO / CMMUA | CMTO / CMTA | CMO / CMA | | LIFT CENTER XCPO/C / XCPA/C | XCLO/C / XCLA/C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.180673 | 0.174533 | 0.355206 | -0.100846 | -0.174533 | 0.000000 | -0.275379 | | 0.558167 | 0.775265 |
| | | 0.034284 | 0.017453 | 0.051737 | -0.007640 | -0.017453 | 0.017453 | -0.007640 | | 0.222834 | 0.485007 |
| 2 | 0.887500 | 0.305825 | 0.174533 | 0.480358 | -0.166954 | -0.174533 | 0.000000 | -0.341487 | | 0.545916 | 0.710901 |
| | | 0.056186 | 0.017453 | 0.073659 | -0.013420 | -0.017453 | 0.017453 | -0.013420 | | 0.238857 | 0.419256 |
| 3 | 0.687500 | 0.529559 | 0.349066 | 0.878625 | -0.313977 | -0.349066 | 0.017453 | -0.645590 | | 0.592904 | 0.754638 |
| | | 0.076014 | 0.017453 | 0.093467 | -0.019453 | -0.017453 | 0.017453 | -0.019453 | | 0.255912 | 0.394857 |
| 4 | 0.275000 | 0.389459 | 0.174533 | 0.563992 | -0.189587 | -0.174533 | 0.000000 | -0.364120 | | 0.486797 | 0.645612 |
| | | 0.089566 | 0.017453 | 0.107019 | -0.023831 | -0.017453 | 0.017453 | -0.023831 | | 0.266070 | 0.385763 |
| TOTAL | | 0.407093 | 0.222529 | 0.629622 | -0.216528 | -0.222529 | 0.004800 | -0.434258 | (APEX) | 0.531889 | 0.697335 |
| | | 0.078903 | 0.017453 | 0.096356 | -0.020516 | -0.017455 | 0.017453 | -0.020516 | (APEX) | 0.260017 | 0.394053 |
| | | | | | -0.114755 | -0.166897 | 0.004800 | -0.276852 | (XMC) | 0.236395 | 0.309927 |
| | | | | | -0.000790 | -0.013090 | 0.013090 | -0.000790 | (XMC) | 0.115563 | 0.175135 |

INDUCED DRAG

| SECTION | Y | CDGO / CDGX / CDGA2 | CDMUO / CDMUX / CDMUA2 | CSO / CSX / CSA2 | CDO / CDX / CDA2 | GAMMAO / GAMMA2 | ALFINO / ALFINA | CTO | CMU |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.975000 | 0.0000000 | 0.0152309 | 0.0008442 | 0.0143867 | 0.0582766 | 0.3064195 | 0.9856133 | 1.0000000 |
| | | 0.0031533 | 0.0030462 | 0.0009947 | 0.0052048 | | | | |
| | | 0.0005994 | 0.0001523 | 0.0002930 | 0.0004577 | 0.0092369 | 0.0430782 | | |
| 2 | 0.887500 | 0.0000000 | 0.0152309 | 0.0021297 | 0.0131011 | 0.0955214 | 0.1378922 | 0.9868989 | 1.0000000 |
| | | 0.0053[illegible]77 | 0.0030462 | 0.0022656 | 0.0061182 | | | | |
| | | 0.0009[illegible]6 | 0.0001523 | 0.0006025 | 0.0005304 | 0.0134614 | 0.0219981 | | |
| 3 | 0.687500 | 0.0195420 | 0.0609235 | 0.0042134 | 0.0762519 | 0.1560372 | 0.1664044 | 0.9237481 | 1.0000000 |
| | | 0.0095343 | 0.0060923 | 0.0039577 | 0.0116690 | | | | |
| | | 0.0013267 | 0.0001523 | 0.0009294 | 0.0005496 | 0.0182760 | 0.0120724 | | |
| 4 | 0.275000 | 0.0000000 | 0.0152309 | 0.0054284 | 0.0098025 | 0.1184530 | 0.1174480 | 0.9901975 | 1.0000000 |
| | | 0.0067973 | 0.0030462 | 0.0050867 | 0.0047568 | | | | |
| | | 0.0015632 | 0.0001523 | 0.0011916 | 0.0005239 | 0.0216792 | 0.0065520 | | |
| TOTAL | | 0.0053740 | 0.0277963 | 0.0044527 | 0.0287176 | | 0.0451802 | 0.9712823 | 0.9999999 |
| | | 0.0071854 | 0.0038839 | 0.0042190 | 0.0068502 | | 0.0101437 | | |
| | | 0.0013771 | 0.0001523 | 0.0010010 | 0.0005285 | | 0.0005049 | | |

* TOTAL AERODYNAMIC COEFFICIENTS *

| | | ALPHA=0 | ALPHA | ALPHA**2 |
|---|---|---|---|---|
| | CCLG | 0.407093 | 0.078903 | |
| | CCLJ | 0.222529 | 0.017453 | |
| ** | CCL | 0.629622 | 0.096356 | |
| | CCDG | 0.0053740 | 0.0071854 | 0.0013771 |
| | CCDJ | 0.0277963 | 0.0038839 | 0.0001525 |
| | CCS | 0.0044527 | 0.0042190 | 0.0010010 |
| | CCD | 0.0287176 | 0.0068502 | 0.0005285 |
| ** | CDITZ | 0.0451802 | 0.0101437 | 0.0005049 |
| ** | CCJ | 0.9999999 | | |
| | CCMG | -0.216528 | -0.020516 | |
| | CCMJ | -0.222529 | -0.017453 | |
| | CCMT | 0.004800 | 0.017453 | |
| | CCM | -0.434258 | -0.020516 | |
| | CXCP | 0.531889 | 0.260017 | |
| | CXCL | 0.697335 | 0.394053 | |
| | CXCPB | 0.236395 | 0.115563 | |
| | CXCLB | 0.309927 | 0.175135 | |
| | CCMGMC | -0.114755 | -0.000790 | |
| | CCMJMC | -0.166897 | -0.013090 | |
| | CCMTMC | 0.004800 | 0.013090 | |
| ** | CCMMC | -0.276852 | -0.000790 | |
| | CLLG | 0.000000 | 0.000000 | |
| | CLLJ | 0.000000 | 0.000000 | |
| * | CLL | 0.000000 | 0.000000 | |
| | CNJ | 0.000000 | | |
| * | CNIMC | 0.000000 | 0.000000 | 0.000000 |
| * | CCY | 0.000000 | 0.000000 | 0.000000 |
| | CBGR | 0.201760 | 0.035823 | |
| | CBGL | 0.201760 | 0.035823 | |
| | CBJR | 0.120264 | 0.008727 | |
| | CBJL | 0.120264 | 0.008727 | |
| | CBR | 0.322024 | 0.044549 | |
| | CBL | 0.322024 | 0.044549 | |
| | CPMBR | 0.511457 | 0.462342 | |
| | CPMBL | 0.511457 | 0.462342 | |

* TABULATED TOTAL COEFFICIENTS FOR COMPOSITE CASE 1 *

| ALPHA | CCL | CCL**2 | CCM(MC) | CLL | CDITZ | CCT | CNI | CN | CCY |
|---|---|---|---|---|---|---|---|---|---|
| -10.000000 | -0.3339359 | 0.1115131 | -0.2689485 | 0.0000000 | -0.0057628 | 1.0057621 | 0.0000000 | 0.0000000 | 0.0000000 |
| -9.000000 | -0.2375801 | 0.0564443 | -0.2697389 | 0.0000000 | -0.0052130 | 1.0052128 | 0.0000000 | 0.0000000 | 0.0000000 |
| -8.000000 | -0.1412245 | 0.0199445 | -0.2705293 | 0.0000000 | -0.0036533 | 1.0036526 | 0.0000000 | 0.0000000 | 0.0000000 |
| -7.000000 | -0.0448685 | 0.0020132 | -0.2713196 | 0.0000000 | -0.0010837 | 1.0010834 | 0.0000000 | 0.0000000 | 0.0000000 |
| -6.000000 | 0.0514875 | 0.0026509 | -0.2721100 | 0.0000000 | 0.0024958 | 0.9975042 | 0.0000000 | 0.0000000 | 0.0000000 |
| -5.000000 | 0.1478431 | 0.0218576 | -0.2729004 | 0.0000000 | 0.0070851 | 0.9929148 | 0.0000000 | 0.0000000 | 0.0000000 |
| -4.000000 | 0.2441989 | 0.0596331 | -0.2736908 | 0.0000000 | 0.0126844 | 0.9873155 | 0.0000000 | 0.0000000 | 0.0000000 |
| -3.000000 | 0.3405547 | 0.1159775 | -0.2744812 | 0.0000000 | 0.0192935 | 0.9807064 | 0.0000000 | 0.0000000 | 0.0000000 |
| -2.000000 | 0.4369105 | 0.1908907 | -0.2752715 | 0.0000000 | 0.0269125 | 0.9730874 | 0.0000000 | 0.0000000 | 0.0000000 |
| -1.000000 | 0.5332662 | 0.2843729 | -0.2760619 | 0.0000000 | 0.0355414 | 0.9644585 | 0.0000000 | 0.0000000 | 0.0000000 |
| 0.000000 | 0.6296220 | 0.3964239 | -0.2768523 | 0.0000000 | 0.0451802 | 0.9548197 | 0.0000000 | 0.0000000 | 0.0000000 |
| 1.000000 | 0.7259778 | 0.5270438 | -0.2776427 | 0.0000000 | 0.0558289 | 0.9441710 | 0.0000000 | 0.0000000 | 0.0000000 |
| 2.000000 | 0.8223336 | 0.6762326 | -0.2784330 | 0.0000000 | 0.0674874 | 0.9325125 | 0.0000000 | 0.0000000 | 0.0000000 |
| 3.000000 | 0.9186894 | 0.8439903 | -0.2792234 | 0.0000000 | 0.0801558 | 0.9198441 | 0.0000000 | 0.0000000 | 0.0000000 |
| 4.000000 | 1.0150452 | 1.0303164 | -0.2800138 | 0.0000000 | 0.0938342 | 0.9061657 | 0.0000000 | 0.0000000 | 0.0000000 |
| 5.000000 | 1.1114006 | 1.2352104 | -0.2808042 | 0.0000000 | 0.1085224 | 0.8914775 | 0.0000000 | 0.0000000 | 0.0000000 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6.000000 | * | 1.2077560 | 1.4586744 | -0.2815945 | 0.0000000 | * | 0.1242206 | 0.8757794 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 7.000000 | * | 1.3041124 | 1.7007084 | -0.2823849 | 0.0000000 | * | 0.1409285 | 0.8590714 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 8.000000 | * | 1.4004679 | 1.9613094 | -0.2831753 | 0.0000000 | * | 0.1586464 | 0.8413535 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 9.000000 | * | 1.4968233 | 2.2404795 | -0.2839656 | 0.0000000 | * | 0.1775741 | 0.8226258 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 10.000000 | * | 1.5931797 | 2.5382214 | -0.2847561 | 0.0000000 | * | 0.1971118 | 0.8028882 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 11.000000 | * | 1.6895351 | 2.8545284 | -0.2855464 | 0.0000000 | * | 0.2178593 | 0.7821407 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 12.000000 | * | 1.7858906 | 3.1894045 | -0.2863368 | 0.0000000 | * | 0.2396168 | 0.7603832 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 13.000000 | * | 1.8822460 | 3.5428495 | -0.2871272 | 0.0000000 | * | 0.2623841 | 0.7376159 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 14.000000 | * | 1.9786024 | 3.9148674 | -0.2879176 | 0.0000000 | * | 0.2861612 | 0.7138388 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 15.000000 | * | 2.0749578 | 4.3054495 | -0.2887079 | 0.0000000 | * | 0.3109483 | 0.6890516 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 16.000000 | * | 2.1713142 | 4.7146053 | -0.2894983 | 0.0000000 | * | 0.3367453 | 0.6632546 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 17.000000 | * | 2.2676697 | 5.1423254 | -0.2902887 | 0.0000000 | * | 0.3635521 | 0.6364478 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 18.000000 | * | 2.3640251 | 5.5886145 | -0.2910790 | 0.0000000 | * | 0.3913689 | 0.6086311 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 19.000000 | * | 2.4603815 | 6.0534765 | -0.2918694 | 0.0000000 | * | 0.4201955 | 0.5798045 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 20.000000 | * | 2.5567369 | 6.5369034 | -0.2926598 | 0.0000000 | * | 0.4500320 | 0.5499679 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 21.000000 | * | 2.6530924 | 7.0388985 | -0.2934502 | 0.0000000 | * | 0.4808784 | 0.5191215 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 22.000000 | * | 2.7494488 | 7.5594683 | -0.2942405 | 0.0000000 | * | 0.5127347 | 0.4872653 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 23.000000 | * | 2.8458042 | 8.0986013 | -0.2950310 | 0.0000000 | * | 0.5456008 | 0.4543991 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 24.000000 | * | 2.9421606 | 8.6563082 | -0.2958213 | 0.0000000 | * | 0.5794769 | 0.4205230 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 25.000000 | * | 3.0385160 | 9.2325792 | -0.2966117 | 0.0000000 | * | 0.6143628 | 0.3856372 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 26.000000 | * | 3.1348715 | 9.8274183 | -0.2974021 | 0.0000000 | * | 0.6502586 | 0.3497413 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 27.000000 | * | 3.2312279 | 10.4408331 | -0.2981924 | 0.0000000 | * | 0.6871643 | 0.3128356 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 28.000000 | * | 3.3275833 | 11.0728102 | -0.2989828 | 0.0000000 | * | 0.7250799 | 0.2749200 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 29.000000 | * | 3.4239388 | 11.7233562 | -0.2997732 | 0.0000000 | * | 0.7640054 | 0.2359945 | * | 0.0000000 | 0.0000000 | 0.0000000 |
| 30.000000 | * | 3.5202951 | 12.3924770 | -0.3005636 | 0.0000000 | * | 0.8039407 | 0.1960592 | * | 0.0000000 | 0.0000000 | 0.0000000 |

```
           ***********************************************
           *  SECOND RUN FOR STABILITY DERIVATIVE CASE  *
           ***********************************************
      *****************************************************
      *  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  1  *
      *****************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING. CLLR =   0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG. CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE  CNR  =  0.000001049
       CNR2 =  0.0000000

SIDE FORCE COEFFICIENT DUE TO YAWING. CY(R) MAY BE CALCULATED AS FOLLOWS

CY(R) = CYR*R + CYR2*R**2

WHERE  CYR  =  0.000000000
       CYR2 =  0.0000000
      *****************************************************
      *  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  2  *
      *****************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING. CLLR =   0.0000436

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG. CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE  CNR  =  0.000000034
       CNR2 =  0.0000000

SIDE FORCE COEFFICIENT DUE TO YAWING. CY(R) MAY BE CALCULATED AS FOLLOWS

CY(R) = CYR*R + CYR2*R**2

WHERE  CYR  =  0.000000000
       CYR2 =  0.0000000
      *****************************************************
      *  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  3  *
      *****************************************************

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING. CLLR =   0.0000344

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG. CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE  CNR  = -0.000000100
       CNR2 =  0.0000000

SIDE FORCE COEFFICIENT DUE TO YAWING. CY(R) MAY BE CALCULATED AS FOLLOWS

CY(R) = CYR*R + CYR2*R**2

WHERE  CYR  =  0.000000000
       CYR2 =  0.0000000
      *****************************************************
      *  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE  4  *
      *****************************************************

ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING. CLLP =  -0.0066949

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING. CN(P) MAY BE CALCULATED AS FOLLOWS

CN(P) = CNP2*P**2

WHERE  CNP2 =   0.0000000

SIDE FORCE COEFFICIENT DUE TO ROLLING. CY(P) MAY BE CALCULATED AS FOLLOWS

CY(P) = CYP2*P**2
```

```
                         WHERE  CYP2 =   0.0000000
        ************************************************************
        *  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE  1  *
        ************************************************************

                         FUNDAMENTAL CASE FACTORS
   A(1)      A(2)      A(3)      A(4)      A(5)      A(6)      A(7)      A(8)      A(9)      A(10)
0.000000 10.000000 10.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000

          LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG. CLQ =  0.089478
 PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG. CMQ = -0.038412
   PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG. CMQMC = -0.016043

            ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING. CLLP =  -0.0066969

  YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING. CN(P) MAY BE CALCULATED AS FOLLOWS
                           CN(P) = CNP*P + CNP2*P**2
                  WHERE    CNP = CNP0 + CNPA*ALPHA
                           CNP0 =  -0.0003531
                           CNPA =  -0.0000601
                           CNP2 =   0.0000000

        SIDE FORCE COEFFICIENT DUE TO ROLLING. CY(P) MAY BE CALCULATED AS FOLLOWS
                           CY(P) = CYP*P + CYP2*P**2
                  WHERE    CYP = CYP0 + CYPA*ALPHA
                           CYP0 =   0.0000000
                           CYPA =   0.0000000
                           CYP2 =   0.0000000

 ROLLING MOMENT COEFF DERIVATIVE DUE TO YAWING ABOUT XCG. CLLR MAY BE CALCULATED AS FOLLOWS
                           CLLR = CLLR0 + CLLRA*ALPHA
                  WHERE     CLLR0 =      0.0007805
                            CLLRA =      0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG. CN(R) MAY BE CALCULATED AS FOLLOWS
                           CN(R) = CNR*R + CNR2*R**2
              WHERE    CNR = CNR0 + CNRA*ALPHA + CNRA2*ALPHA**2
                           CNR0 =     -0.0000130
                           CNRA =     -0.0000025
                          CNRA2 =      0.0000010
              AND     CNR2 = CNR20 + CNR2A*ALPHA + CNR2A2*ALPHA**2
                          CNR20 =      0.0000000
                          CNR2A =      0.0000000
                         CNR2A2 =      0.0000000

 SIDE FORCE COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG. CY(R) MAY BE CALCULATED AS FOLLOWS
                           CY(R) = CYR*R + CYR2*R**2
              WHERE    CYR = CYR0 + CYRA*ALPHA + CYRA2*ALPHA**2
                           CYR0 =      0.0000000
                           CYRA =      0.0000000
                          CYRA2 =      0.0000000
              AND     CYR2 = CYR20 + CYR2A*ALPHA + CYR2A2*ALPHA**2
                          CYR20 =      0.0000000
                          CYR2A =      0.0000000
                         CYR2A2 =      0.0000000
        ************************************************************
        * VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK *
        ************************************************************
```

| ALPHA | | CNP | CNP2 | | CYP | CYP2 | | CLLR | CNR | CNR2 |
|---|---|---|---|---|---|---|---|---|---|---|
| -10.000000 | * | 0.0002479 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | -0.0003880 | 0.0001273 | 0.0000000 |
| -9.000000 | * | 0.0001878 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | -0.0002712 | 0.0001038 | 0.0000000 |
| -8.000000 | * | 0.0001277 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | -0.0001543 | 0.0000824 | 0.0000000 |
| -7.000000 | * | 0.0000676 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | -0.0000375 | 0.0000632 | 0.0000000 |
| -6.000000 | * | 0.0000075 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0000794 | 0.0000460 | 0.0000000 |
| -5.000000 | * | -0.0000526 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0001962 | 0.0000309 | 0.0000000 |
| -4.000000 | * | -0.0001127 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0003131 | 0.0000179 | 0.0000000 |
| -3.000000 | * | -0.0001728 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0004299 | 0.0000070 | 0.0000000 |
| -2.000000 | * | -0.0002329 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0005468 | -0.0000017 | 0.0000000 |
| -1.000000 | * | -0.0002930 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0006636 | -0.0000084 | 0.0000000 |
| 0.000000 | * | -0.0003531 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0007805 | -0.0000130 | 0.0000000 |
| 1.000000 | * | -0.0004132 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0008973 | -0.0000155 | 0.0000000 |
| 2.000000 | * | -0.0004733 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0010142 | -0.0000159 | 0.0000000 |
| 3.000000 | * | -0.0005334 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0011310 | -0.0000142 | 0.0000000 |
| 4.000000 | * | -0.0005935 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0012479 | -0.0000104 | 0.0000000 |
| 5.000000 | * | -0.0006536 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0013647 | -0.0000045 | 0.0000000 |
| 6.000000 | * | -0.0007137 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0014815 | 0.0000035 | 0.0000000 |
| 7.000000 | * | -0.0007738 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0015984 | 0.0000136 | 0.0000000 |
| 8.000000 | * | -0.0008339 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0017152 | 0.0000258 | 0.0000000 |
| 9.000000 | * | -0.0008940 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0018321 | 0.0000401 | 0.0000000 |
| 10.000000 | * | -0.0009541 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0019489 | 0.0000565 | 0.0000000 |
| 11.000000 | * | -0.0010142 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0020658 | 0.0000750 | 0.0000000 |
| 12.000000 | * | -0.0010743 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0021826 | 0.0000956 | 0.0000000 |
| 13.000000 | * | -0.0011344 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0022995 | 0.0001182 | 0.0000000 |
| 14.000000 | * | -0.0011945 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0024163 | 0.0001430 | 0.0000000 |
| 15.000000 | * | -0.0012546 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0025332 | 0.0001699 | 0.0000000 |
| 16.000000 | * | -0.0013147 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0026500 | 0.0001989 | 0.0000000 |
| 17.000000 | * | -0.0013748 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0027669 | 0.0002299 | 0.0000000 |
| 18.000000 | * | -0.0014349 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0028837 | 0.0002631 | 0.0000000 |
| 19.000000 | * | -0.0014950 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0030006 | 0.0002984 | 0.0000000 |
| 20.000000 | * | -0.0015551 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0031174 | 0.0003358 | 0.0000000 |
| 21.000000 | * | -0.0016152 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0032343 | 0.0003752 | 0.0000000 |
| 22.000000 | * | -0.0016753 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0033511 | 0.0004168 | 0.0000000 |
| 23.000000 | * | -0.0017354 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0034680 | 0.0004604 | 0.0000000 |
| 24.000000 | * | -0.0017955 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0035848 | 0.0005062 | 0.0000000 |
| 25.000000 | * | -0.0018556 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0037017 | 0.0005541 | 0.0000000 |
| 26.000000 | * | -0.0019157 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0038185 | 0.0006040 | 0.0000000 |
| 27.000000 | * | -0.0019758 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0039353 | 0.0006561 | 0.0000000 |
| 28.000000 | * | -0.0020359 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0040522 | 0.0007102 | 0.0000000 |
| 29.000000 | * | -0.0020960 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0041690 | 0.0007665 | 0.0000000 |
| 30.000000 | * | -0.0021561 | 0.0000000 | * | 0.0000000 | 0.0000000 | * | 0.0042859 | 0.0008248 | 0.0000000 |

```
        ************************************************************
        *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
        ************************************************************
```


```
        ************************************************************
        *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
        ************************************************************
```

## JETFLAP INPUT DATA FILE TAPER.DAT

```
TAPERED SWEPT WING, AR=8.0, SWEEP ANGLE 45, 10X10 W/SEMI-CIRCLE SPACING
50.0000   20.000    0.0        10.43      10.43
1001000001020000
.993844   .969372   .921032   .850012   .758062   .647446   .520888   .381504
.232726   .078217
01010101010101010101
10
.0        .024472   .095492   .206107   .345492   .5000     .654508   .793893
.904508   .975528
8.0       45.0      0.45
9
```

# PROGRAM OUTPUT DATA FOR TAPER.DAT

```
                    ***************************************
                    *  EVD  JET - WING  COMPUTER  PROGRAM  *
                    ***************************************

     TAPERED SWEPT WING, AR=8.0, SWEEP ANDGLE 45, 10X10 W/SEMI-CIRCLE SPACING

                            USED            INPUT
                 AREA =   0.500000        50.000000
                 SPAN =   2.000000        20.000000
                 CREF =   0.261990         0.000000
                  XMC =   1.043000        10.430000
                 CMAC =   0.261794         2.617941
               ARATIO =   8.000000         8.000000
                  XCG =   1.043000        10.430000

                NROWS =  10           10
               NCASES =   1            1
                ISYMM =   0            0
               IPRINT =   0            0
               JETFLG =   1            1
               IGTYPE =   2            2
               IHINGE =   0            0

             NUMBER OF WING ELEMENTS = 100
             NUMBER OF JET ELEMENTS  =   0
             TOTAL NUMBER OF ELEMENTS = 100
     ******************************************************************************
     *  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  1  *
     ******************************************************************************
*** SECTION  1 ***  Y =  0.993844  DELTA =  0.006156  XLEAD =  1.040966  XTRAIL
=  1.197306  CHORD =  0.156340  TANLE =  1.047412
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  1.040966  1.044792  1.055895  1.073189  1.094980  1.119136  1.143292  1.165083  1.182377  1.193480
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  2 ***  Y =  0.969372  DELTA =  0.018316  XLEAD =  1.015334  XTRAIL =  1.176315  CHORD =  0.160981  TANLE =  1.047412
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  1.015334  1.019273  1.030706  1.048513  1.070951  1.095824  1.120697  1.143135  1.160942  1.172373
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  3 ***  Y =  0.921032  DELTA =  0.030024  XLEAD =  0.964702  XTRAIL =  1.134851  CHORD =  0.170149  TANLE =  1.047415
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.964702  0.968865  0.980949  0.999770  1.023487  1.049776  1.076066  1.099782  1.118603  1.130687
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  4 ***  Y =  0.850012  DELTA =  0.040996  XLEAD =  0.890314  XTRAIL =  1.073933  CHORD =  0.183618  TANLE =  1.047413
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.890314  0.894808  0.907848  0.928159  0.953753  0.982123  1.010494  1.036088  1.056399  1.069439
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  5 ***  Y =  0.758062  DELTA =  0.050956  XLEAD =  0.794005  XTRAIL =  0.995062  CHORD =  0.201057  TANLE =  1.047413
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.794005  0.798925  0.813204  0.835444  0.863468  0.894533  0.925598  0.953622  0.975862  0.990142
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  6 ***  Y =  0.647666  DELTA =  0.059662  XLEAD =  0.678144  XTRAIL =  0.900180  CHORD =  0.222036  TANLE =  1.047414
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.678144  0.683577  0.699346  0.723907  0.754856  0.789162  0.823468  0.854417  0.878977  0.894746
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  7 ***  Y =  0.520888  DELTA =  0.066896  XLEAD =  0.545585  XTRAIL
=  0.791624  CHORD =  0.246038  TANLE =  1.047416
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.545585  0.551606  0.569080  0.596295  0.630590  0.668604  0.706619  0.740913  0.768129  0.785603
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  8 ***  Y =  0.381504  DELTA =  0.072408  XLEAD =  0.399593  XTRAIL =  0.672066  CHORD =  0.272473  TANLE =  1.047416
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.399593  0.406260  0.425612  0.455751  0.493730  0.535829  0.577929  0.615907  0.646047  0.665398
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
      THIS ROW HAS NO JET
*** SECTION  9 ***  Y =  0.232726  DELTA =  0.076290  XLEAD =  0.243760  XTRAIL =  0.544450  CHORD =  0.300690  TANLE =  1.047416
WING ELEMENTS    NW = 10      TWIST =  0.000000      ML =  0.000000      THETA S =  0.000000
         XB  0.000000  0.024472  0.095492  0.206107  0.345492  0.500000  0.654508  0.793893  0.904508  0.975528
         XI  0.243760  0.251119  0.272474  0.305735  0.347646  0.394105  0.440564  0.482476  0.515737  0.537092
        DEL  0.024472  0.071020  0.110615  0.139385  0.154508  0.154508  0.139385  0.110615  0.071020  0.024472
        EPS  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
       BETA  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
       TYPE  20        10        10        10        10        10        10        10        10        10
```

THIS ROW HAS NO JET

*** SECTION 10 *** Y = 0.078217 DELTA = 0.078219 XLEAD = 0.081926 XTRAIL = 0.411919 CHORD = 0.329993 TANLE = 1.047414

WING ELEMENTS NW = 10 TWIST = 0.000000 HL = 0.000000 THETA S = 0.000000

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XB | 0.000000 | 0.024472 | 0.095492 | 0.206107 | 0.345492 | 0.500000 | 0.654508 | 0.793893 | 0.904508 | 0.975528 |
| XI | 0.081926 | 0.090001 | 0.113437 | 0.149939 | 0.195936 | 0.246922 | 0.297909 | 0.343905 | 0.380407 | 0.403843 |
| DEL | 0.024472 | 0.071020 | 0.110615 | 0.139385 | 0.154508 | 0.154508 | 0.139385 | 0.110615 | 0.071020 | 0.024472 |
| EPS | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| BETA | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| TYPE | 20 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

THIS ROW HAS NO JET

***************************************

* SECTIONAL JET BLOWING COEFFICIENTS

***************************************

| ROW | CMU |
|---|---|
| 1 | 0.000000 |
| 2 | 0.000000 |
| 3 | 0.000000 |
| 4 | 0.000000 |
| 5 | 0.000000 |
| 6 | 0.000000 |
| 7 | 0.000000 |
| 8 | 0.000000 |
| 9 | 0.000000 |
| 10 | 0.000000 |

****************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

****************************************************

SECTION 1 Y = 0.993844 CHORD = 0.156340

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.000000 | 0.565735 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.024472 | 0.244147 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 3 | 0.095492 | 0.050298 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 4 | 0.206107 | 0.020919 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 5 | 0.345492 | 0.011226 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 6 | 0.500000 | 0.006413 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 7 | 0.654508 | 0.003731 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 8 | 0.793893 | 0.001994 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 9 | 0.904508 | 0.000927 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 10 | 0.975528 | 0.000486 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| | | |
|---|---|---|
| 1 | 0.004894 | 0.816745 |
| 2 | 0.009789 | 0.543132 |
| 3 | 0.014683 | 0.407101 |
| 4 | 0.019578 | 0.315266 |
| 5 | 0.024472 | 0.244147 |

****************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

****************************************************

SECTION 2 Y = 0.969372 CHORD = 0.160981

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 0.000000 | 0.570113 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 12 | 0.024472 | 0.320536 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 13 | 0.095492 | 0.126056 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 14 | 0.206107 | 0.064482 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 15 | 0.345492 | 0.033466 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 16 | 0.500000 | 0.017851 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 17 | 0.654508 | 0.010013 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 18 | 0.793893 | 0.005811 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 19 | 0.904508 | 0.003360 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 20 | 0.975528 | 0.001583 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| | | |
|---|---|---|
| 1 | 0.004894 | 0.837965 |
| 2 | 0.009789 | 0.577136 |
| 3 | 0.014683 | 0.454951 |
| 4 | 0.019578 | 0.377306 |
| 5 | 0.024472 | 0.320536 |

****************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

****************************************************

SECTION 3 Y = 0.921032 CHORD = 0.170149

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 21 | 0.000000 | 0.583867 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 22 | 0.024472 | 0.341116 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 23 | 0.095492 | 0.149001 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 24 | 0.206107 | 0.084181 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25 | 0.345492 | 0.055031 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 26 | 0.500000 | 0.036803 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 27 | 0.654508 | 0.023636 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 28 | 0.793893 | 0.014871 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 29 | 0.904508 | 0.008942 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 30 | 0.975528 | 0.004208 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| | | |
|---|---|---|
| 1 | 0.004894 | 0.860751 |
| 2 | 0.009789 | 0.596198 |
| 3 | 0.014683 | 0.473635 |
| 4 | 0.019578 | 0.396686 |
| 5 | 0.024472 | 0.341116 |

****************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

****************************************************

SECTION 4 Y = 0.850012 CHORD = 0.183618

| WING I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 0.000000 | 0.582767 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 32 | 0.024472 | 0.344836 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 33 | 0.095492 | 0.156110 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 34 | 0.206107 | 0.090613 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 35 | 0.345492 | 0.059662 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 36 | 0.500000 | 0.043749 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 37 | 0.654508 | 0.031802 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 38 | 0.793893 | 0.022199 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 39 | 0.904508 | 0.014177 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 40 | 0.975528 | 0.006920 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| | | |
|---|---|---|
| 1 | 0.004894 | 0.859974 |
| 2 | 0.009789 | 0.596803 |
| 3 | 0.014683 | 0.475350 |
| 4 | 0.019578 | 0.399622 |
| 5 | 0.024472 | 0.344836 |

****************************************************

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

****************************************************
SECTION 5 Y = 0.758062 CHORD = 0.201057

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 41 | 0.000000 | 0.571170 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 42 | 0.024472 | 0.339773 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 43 | 0.095492 | 0.156167 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 44 | 0.206107 | 0.091835 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 45 | 0.345492 | 0.060084 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 46 | 0.500000 | 0.044228 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 47 | 0.654508 | 0.033507 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 48 | 0.793893 | 0.024279 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 49 | 0.904508 | 0.015937 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 50 | 0.975528 | 0.007778 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.004894 | 0.863248 |
| 2 | 0.009789 | 0.585663 |
| 3 | 0.014683 | 0.466981 |
| 4 | 0.019578 | 0.392920 |
| 5 | 0.024472 | 0.339773 |

****************************************************
* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *
****************************************************
SECTION 6 Y = 0.647446 CHORD = 0.222036

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 51 | 0.000000 | 0.552733 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 52 | 0.024472 | 0.329515 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 53 | 0.095492 | 0.152388 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 54 | 0.206107 | 0.090136 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 55 | 0.345492 | 0.058818 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 56 | 0.500000 | 0.043056 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 57 | 0.654508 | 0.033043 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 58 | 0.793893 | 0.024299 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 59 | 0.904508 | 0.016226 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 60 | 0.975528 | 0.007943 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.004894 | 0.816171 |
| 2 | 0.009789 | 0.567042 |
| 3 | 0.014683 | 0.452333 |
| 4 | 0.019578 | 0.380804 |
| 5 | 0.024472 | 0.329515 |

****************************************************
* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *
****************************************************
SECTION 7 Y = 0.520888 CHORD = 0.246038

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 61 | 0.000000 | 0.528120 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 62 | 0.024472 | 0.315010 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 63 | 0.095492 | 0.145907 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 64 | 0.206107 | 0.086468 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 65 | 0.345492 | 0.056458 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 66 | 0.500000 | 0.041378 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 67 | 0.654508 | 0.031972 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 68 | 0.793893 | 0.023680 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 69 | 0.904508 | 0.015679 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 70 | 0.975528 | 0.007727 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.004894 | 0.779860 |
| 2 | 0.009789 | 0.541859 |
| 3 | 0.014683 | 0.432291 |
| 4 | 0.019578 | 0.363991 |
| 5 | 0.024472 | 0.315010 |

****************************************************
* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *
****************************************************
SECTION 8 Y = 0.381504 CHORD = 0.272473

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 71 | 0.000000 | 0.495801 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 72 | 0.024472 | 0.295628 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 73 | 0.095492 | 0.136824 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 74 | 0.206107 | 0.081121 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 75 | 0.345492 | 0.053235 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 76 | 0.500000 | 0.039407 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 77 | 0.654508 | 0.030708 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 78 | 0.793893 | 0.022843 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 79 | 0.904508 | 0.015165 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 80 | 0.975528 | 0.007489 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.004894 | 0.732116 |
| 2 | 0.009789 | 0.508657 |
| 3 | 0.014683 | 0.405773 |
| 4 | 0.019578 | 0.341623 |
| 5 | 0.024472 | 0.295628 |

****************************************************
* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *
****************************************************
SECTION 9 Y = 0.232726 CHORD = 0.300690

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 81 | 0.000000 | 0.448660 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 82 | 0.024472 | 0.267585 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 83 | 0.095492 | 0.124018 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 84 | 0.206107 | 0.074036 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 85 | 0.345492 | 0.049507 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 86 | 0.500000 | 0.037572 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 87 | 0.654508 | 0.029646 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 88 | 0.793893 | 0.022142 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 89 | 0.904508 | 0.014743 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 90 | 0.975528 | 0.007298 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| I | XB | CASE 1 |
|---|---|---|
| 1 | 0.004894 | 0.662517 |
| 2 | 0.009789 | 0.460320 |
| 3 | 0.014683 | 0.367232 |
| 4 | 0.019578 | 0.309196 |
| 5 | 0.024472 | 0.267585 |

****************************************************
* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *
****************************************************
SECTION 10 Y = 0.078217 CHORD = 0.329993

| WING | I | XB | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | CASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 91 | 0.000000 | 0.348952 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 92 | 0.024472 | 0.210196 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 93 | 0.095492 | 0.100887 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 94 | 0.206107 | 0.064371 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| | 95 | 0.345492 | 0.047291 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 96 | 0.500000 | 0.039165 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 97 | 0.654508 | 0.032400 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 98 | 0.793893 | 0.024758 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 99 | 0.904508 | 0.016676 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 100 | 0.975528 | 0.008296 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

DETAILED LEADING EDGE LOADING

| | | |
|---|---|---|
| 1 | 0.004894 | 0.515699 |
| 2 | 0.009789 | 0.358852 |
| 3 | 0.014683 | 0.286867 |
| 4 | 0.019578 | 0.242142 |
| 5 | 0.024472 | 0.210196 |

* SPANWISE LOADING FOR FUNDAMENTAL CASE 1*

| SECTION | Y | CLG | CLMU | CL | * | CDG | CDMU | CS | CD | CMU | GAMMA | ALFIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ........... LIFT ........... | | | | .................... INDUCED DRAG | | | | | | |
| 1 | 0.993844 | 0.036230 | 0.000000 | 0.036230 | * | 0.0006323 | 0.0000000 | 0.0013670 | -0.0007347 | 0.0000000 | 0.0028321 | 0.0570169 |
| 2 | 0.969372 | 0.059019 | 0.000000 | 0.059019 | * | 0.0010301 | 0.0000000 | 0.0013883 | -0.0003582 | 0.0000000 | 0.0047505 | 0.0255027 |
| 3 | 0.921032 | 0.074748 | 0.000000 | 0.074748 | * | 0.0013046 | 0.0000000 | 0.0014560 | -0.0001514 | 0.0000000 | 0.0063592 | 0.0127912 |
| 4 | 0.850012 | 0.080822 | 0.000000 | 0.080822 | * | 0.0014106 | 0.0000000 | 0.0014505 | -0.0000398 | 0.0000000 | 0.0074202 | 0.0086195 |
| 5 | 0.758062 | 0.081304 | 0.000000 | 0.081304 | * | 0.0014190 | 0.0000000 | 0.0013934 | 0.0000256 | 0.0000000 | 0.0081734 | 0.0061406 |
| 6 | 0.647446 | 0.079389 | 0.000000 | 0.079389 | * | 0.0013856 | 0.0000000 | 0.0013049 | 0.0000807 | 0.0000000 | 0.0088136 | 0.0050705 |
| 7 | 0.520888 | 0.076165 | 0.000000 | 0.076165 | * | 0.0013293 | 0.0000000 | 0.0011913 | 0.0001380 | 0.0000000 | 0.0093697 | 0.0047486 |
| 8 | 0.381504 | 0.071829 | 0.000000 | 0.071829 | * | 0.0012536 | 0.0000000 | 0.0010499 | 0.0002037 | 0.0000000 | 0.0097857 | 0.0047268 |
| 9 | 0.232726 | 0.066165 | 0.000000 | 0.066165 | * | 0.0011548 | 0.0000000 | 0.0008598 | 0.0002950 | 0.0000000 | 0.0099475 | 0.0046069 |
| 10 | 0.078217 | 0.058551 | 0.000000 | 0.058551 | * | 0.0010219 | 0.0000000 | 0.0005201 | 0.0005018 | 0.0000000 | 0.0096608 | 0.0031973 |
| TOTAL | | 0.070281 | 0.000000 | 0.070281 | * | 0.0012266 | 0.0000000 | 0.0010305 | 0.0001961 | 0.0000000 | | 0.0002059 |

| SECTION | Y | CMG | CMMU | CMT | CM | * | * | XCP/C | XCL/C |
|---|---|---|---|---|---|---|---|---|---|
| | | .......... PITCHING MOMENT ........... | | | | | | ... LIFT CENTER ... | |
| 1 | 0.993844 | -0.003325 | 0.000000 | 0.000000 | -0.003325 | * | * | 0.091763 | 0.091763 |
| 2 | 0.969372 | -0.008643 | 0.000000 | 0.000000 | -0.008643 | * | * | 0.146453 | 0.146453 |
| 3 | 0.921032 | -0.014778 | 0.000000 | 0.000000 | -0.014778 | * | * | 0.197701 | 0.197701 |
| 4 | 0.850012 | -0.017849 | 0.000000 | 0.000000 | -0.017849 | * | * | 0.220845 | 0.220845 |
| 5 | 0.758062 | -0.018479 | 0.000000 | 0.000000 | -0.018479 | * | * | 0.227287 | 0.227287 |
| 6 | 0.647446 | -0.018190 | 0.000000 | 0.000000 | -0.018190 | * | * | 0.229127 | 0.229127 |
| 7 | 0.520888 | -0.017544 | 0.000000 | 0.000000 | -0.017544 | * | * | 0.230347 | 0.230347 |
| 8 | 0.381504 | -0.016696 | 0.000000 | 0.000000 | -0.016696 | * | * | 0.232438 | 0.232438 |
| 9 | 0.232726 | -0.015772 | 0.000000 | 0.000000 | -0.015772 | * | * | 0.238368 | 0.238368 |
| 10 | 0.078217 | -0.015890 | 0.000000 | 0.000000 | -0.015890 | * | * | 0.271392 | 0.271392 |
| TOTAL | | -0.145988 | 0.000000 | 0.000000 | -0.145988 | (APEX) | | 2.077202 | 2.077202 (X/CREF) |
| | | 0.133806 | 0.000000 | 0.000000 | 0.133806 | (XMC) | | 0.544205 | 0.544205 (X/B/2) |

* TOTAL AERODYNAMIC COEFFICIENTS *

| | CASE 1 | CASE 2 | CASE 3 | CASE 4 | CASE 5 | CASE 6 | ASE 7 | CASE 8 | CASE 9 | CASE 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CCLG | 0.0702811 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCLJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** CCL | 0.0702811 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCDG | 0.0012266 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCDJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCS | 0.0010305 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCD | 0.0001961 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** CDITS | 0.0002059 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** CCJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMG | -0.1459882 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMT | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCM | -0.1459882 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CXCP | 2.0772021 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CXCL | 2.0772021 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CXCPB | 0.5442055 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CXCLB | 0.5442055 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMGMC | 0.1338062 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMJMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CCMTMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| ** CCMMC | 0.1338062 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CLLG | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CLLJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * CLL | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CNJ | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * CNIMC | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| * CCY | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBGR | 0.0324014 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBGL | 0.0324014 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBJR | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBJL | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBR | 0.0324014 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CBL | 0.0324014 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CPMBR | 0.4610249 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |
| CPMBL | 0.4610249 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 | 0.0000000 |

* THE PROGRAM HAS REACHED NORMAL TERMINATION *

* THE PROGRAM HAS REACHED NORMAL TERMINATION *

# PROGRAM JETFLAP LISTING

```
      PROGRAM JETFLAP
C   *** VERSION 3.0 MODIFIED BY J.A. CAMPBELL (JUL88) ***
C   *** PROGRAM REVISED TO RUN UNDER FORTRAN 77 ON THE MICROVAX/2000 ***
C       FINAL UPDATES MADE 14 SEP 88 - (JAC)
C
C   SCRATCH FILES ADDED AND "FIND(" STATEMENTS HAVE BEEN COMMENTED OUT
C   DATA INPUT IS READ BY OBTAINING AN AVAILABLE LOGICAL UNIT NUMBER (LUN)
C   INSTEAD OF ASSIGNING A READ DEVICE AS IS DONE ON THE IBM SYSTEM
C
C   OUTPUT VALIDATED WITH HISTORICAL DATA - RESULTS OF DOUGLAS COMPANY,
C   SODERMAN THESIS AND THE AE-4501 CLASS PROJECT OF S.M. WHITE (MAY 85)
C
C
C   *** VERSION 2.0 UNDER REVISION  J.A. CAMPBELL (FEB 88) ***
C   COMPILED USING "FORTVS JTFLAP(LVL(66))" ON IBM WITH NO ERRORS 5/20/88
C   UPDATED EQN FOR CMG(K) IN SUBR SLOAD (TAPE VERSION DIFFERENT) 5/31/88
C   OUTPUT VALIDATED WITH HISTORICAL DATA - RESULTS OF DOUGLAS CONTRACT &
C   SODERMAN (NPS THESIS)
C
C***********************************************************************
C******************   EVD JET-WING COMPUTER PROGRAM   *******************
C***********************************************************************
C
C
C
C   THE DEVELOPMENT OF THE ELEMENTARY VORTEX DISTRIBUTION (EVD) JET-WING
C   COMPUTER PROGRAM FOR DETERMINING THE AERODYNAMIC CHARACTERISTICS OF
C   ARBITRARY JET FLAPPED WINGS WAS PERFORMED BY THE DOUGLAS AIRCRAFT
C   COMPANY, V/STOL TECHNOLOGY GROUP - AERODYNAMICS, OF THE MCDONNELL
C   DOUGLAS CORPORATION.    ORIGINALLY DEVELOPED UNDER THE SPONSORSHIP OF
C   THE INDEPENDENT RESEARCH AND DEVELOPMENT PROGRAM OF MCDONNELL
C   DOUGLAS, THE DOUGLAS EVD JET WING LIFTING SURFACE THEORY HAS BEEN
C   THE SUBJECT OF EXTENSIONS AND IMPROVEMENTS WHICH HAVE BEEN
C   ACCOMPLISHED UNDER OFFICE OF NAVAL RESEARCH CONTRACT N00014-71-C-0250
C   (T. L. WILSON - PROJECT ENGINEER).    WORK LEADING TO THE PRESENT
C   COMPUTER PROGRAM, WHICH INCORPORATES SEVERAL FEATURES ORIGINALLY
C   DEVELOPED BY THE DOUGLAS AIRCRAFT COMPANY, WAS ALSO CONDUCTED
C   UNDER THE SPONSORSHIP OF THIS CONTRACT.
C
C   IN SUMMARY, THE EVD JET WING COMPUTER PROGRAM WILL PROVIDE,
C   FOR ARBITRARY PLANFORMS, THE FOLLOWING -
C           1.   SPANWISE AND CHORDWISE LOADING
C           2.   SPANWISE VARIATION OF INDUCED DRAG
C           3.   A CAPABILITY TO INVESTIGATE THE EFFECTS OF -
C                A.   PART SPAN FLAPS
C                B.   PART SPAN BLOWING
C                C.   ROLLING, YAWING, PITCHING AND SIDESLIP
C           4.   TOTAL LIFT AND INDUCED DRAG (TREFFTZ PLANE METHOD),
C                PITCHING, YAWING AND ROLLING MOMENTS, ETC.
C
C   COMPLETE DOCUMENTATION OF THE EVD JET WING LIFTING SURFACE THEORY
C   AND ASSOCIATED COMPUTER PROGRAM ARE CONTAINED IN DOUGLAS REPORT
C
C               J5519 -- A THEORETICAL METHOD FOR CALCULATING THE
C         AERODYNAMIC CHARACTERISTICS OF ARBITRARY JET FLAP WINGS
C
C                                VOLUME I
C                THE ELEMENTARY VORTEX DISTRIBUTION JET-WING
C                        LIFTING SURFACE THEORY
C
C                                VOLUME II
C            EVD JET-WING COMPUTER PROGRAM USERS MANUAL
C
C
C
      INTEGER*4 LUN
      INTEGER*2 INFILE_SIZE,IOFILE_SIZE
      INTEGER   STATUS,NANS
      CHARACTER*20 INFILE,OUTFILE
      LOGICAL   EXIST
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(20)
      COMMON/JOHN/  AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/B(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/DERIV/U0(40),CLQ,CMQ,CMQMC
      COMMON/INDAT/LUN
C      DATA CHECK/4H9   ./
      DATA CHECK/'9   '/
C
C
C     DEFINE FILE 1(1000,1200,U,NEXT)
C
C FOLLOWING LINES FOR SCRATCH FILES ADDED BY J.A. CAMPBELL (JUL88)
C
      OPEN (UNIT=1,
     2       FILE= 'JTFLAP1.DAT',
     2       ORGANIZATION= 'SEQUENTIAL',
```

```
     2        ACCESS= 'DIRECT',
     2        RECORDTYPE= 'FIXED',
     2        FORM= 'UNFORMATTED',
     2        RECL= 600,
     2        ASSOCIATEVARIABLE= NEXT,
     2        STATUS= 'SCRATCH')
C OPEN SCRATCH FILE FOR MATRIX INPUT TO SOLN ROUTINE
      OPEN (UNIT=2,
     2        FILE= 'JTFLAP2.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'UNFORMATTED',
     2        STATUS= 'SCRATCH')
C OPEN SCRATCH FILE FOR TEMP STORAGE DURING MATRIX SOLN
      OPEN (UNIT=3,
     2        FILE= 'JTFLAP3.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'UNFORMATTED',
     2        STATUS= 'SCRATCH')
C OPEN SCRATCH FILE FOR TEMP STORAGE DURING MATRIX SOLN
      OPEN (UNIT=4,
     2        FILE= 'JTFLAP4.DAT',
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'UNFORMATTED',
     2        STATUS= 'SCRATCH')
C
C  CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THEN PRINT HEADER
      ISTAT = LIB$ERASE_PAGE (1,1)
      PRINT *
      PRINT *, ' PROGRAM JETFLAP : VERSION 3 : 31 JULY 88 '
      PRINT *
      PRINT *, ' THE ELEMENTARY VORTEX DISTRIBUTION (EVD) JET-WING'
      PRINT *, ' COMPUTER PROGRAM FOR DETERMINING THE AERODYNAMIC'
      PRINT *, ' CHARACTERISTICS OF ARBITRARY JET FLAPPED WINGS'
      PRINT *
      PRINT *
C
C ROUTINE TO PROVIDE NAME FOR AND OPEN INPUT DATA FILE
    5 STATUS = LIB$GET_INPUT (INFILE,          ! The input file
     2          ' ENTER THE DATA FILE NAME:   ',  ! Prompt
     2                          INFILE_SIZE)    ! Filename size
      IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL (STATUS))
C  CHECK TO SEE IF THE FILE EXISTS BEFORE TRYING TO ACCESS IT
      IF (INFILE .EQ. '999') GO TO 110
      INQUIRE (FILE = INFILE (1:INFILE_SIZE), EXIST = EXIST)
      IF (.NOT. EXIST) THEN
         PRINT *
         PRINT *, ' THAT FILE NAME DOES NOT EXIST.'
         PRINT *, '     (ENTER 999 TO EXIT).'
         PRINT *
         GO TO 5
      END IF
C  GET A FREE LOGICAL UNIT NUMBER
      STATUS = LIB$GET_LUN (LUN)
      IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL (STATUS))
C OPEN FILE FOR DATA FILE INPUT
      OPEN (UNIT=LUN,
     2        FILE= INFILE (1:INFILE_SIZE),
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'OLD')
C
C SEND OUTPUT TO SCREEN OR FILE
      CALL CLRSCRN
      PRINT *
      PRINT *, '==> SEND THE RESULTS TO THE SCREEN OR A FILE?'
    6 PRINT *, '           ENTER (S OR F)'
      READ (5,'(A1)') ANS
      IF (ANS.EQ.'F') THEN
        PRINT *
    7   PRINT *
        PRINT *, ' (ENTER 999 TO EXIT.)'
        STATUS = LIB$GET_INPUT (OUTFILE,           ! The OUTPUT file
     2     ' ENTER NAME OF OUTPUT FILE TO CREATE:  ',  ! Prompt
     2                          IOFILE_SIZE)        ! Filename size
C  CHECK TO SEE IF THE FILE EXISTS BEFORE CREATING IT
        IF (OUTFILE .EQ. '999') GO TO 110
        INQUIRE (FILE = OUTFILE (1:IOFILE_SIZE), EXIST = EXIST)
        IF ( EXIST ) THEN
           PRINT *
           PRINT *, ' THAT FILE ALREADY EXISTS.'
           WRITE (6,1005)
           PRINT *, ' (OR ENTER 999 TO RETURN TO EXIT OPTION).'
           PRINT *
    d      CALL QUERY (NANS)
        ELSE
           GO TO 9
      END IF
      IF (NANS .EQ. 1) THEN
         GO TO 9
      ELSE IF(NANS .EQ. 2)THEN
         GO TO 7
      ELSE IF(NANS .EQ. 999)THEN
         GO TO 110
```

```
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 6
      END IF
C OPEN FILE FOR RESULTS FROM PROGRAM JETFLAP
    9    PRINT *, '  PROCESSING BEGINS . . . '
         PRINT *
         PRINT *, '  DATA BEING WRITTEN TO FILE ',OUTFILE
         PRINT *, '  FILE WILL HAVE SUFFIX .DAT'
         OPEN (UNIT=6,FILE=OUTFILE,STATUS='UNKNOWN')
      ELSE IF (ANS.EQ.'S') THEN
         GO TO 10
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 6
      END IF
      PRINT *
 1005 FORMAT (1X,' DO YOU WISH TO OVERWRITE THIS FILE? 1 = YES;2 = NO')
 1010 FORMAT(A4)
C   READ THE TITLE FOR THIS CASE
   10 READ(LUN, 20, END=100 ) TITLE
   20 FORMAT(20A4)
C
C   READ GENERAL GEOMETRY CONTROL DATA
   30 READ(LUN, 40 ) AREA,SPAN,CREF,XMC,XCG
   40 FORMAT(5F10.6)
      READ(LUN,41) NROWS,NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE,IDERIV
   41 FORMAT(10I2)
      ARE = AREA
      SPA = SPAN
      CRE = CREF
      XM  = XMC
      XC  = XCG
      NRO = NROWS
      NC  = NCASES
      ISY = ISYMM
      IPR = IPRINT
      JET = JETFLG
      IGT = IGTYPE
      IHI = IHINGE
C
C   FIND OUT WHICH TYPE OF RUN IS REQUIRED
      IF(IDERIV .NE. 0) GO TO 60
C
C   A REGULAR RUN WILL BE EXECUTED
   50 CALL APPLY1
      GO TO ( 60 , 70 , 100 , 120 ), IR
C
C   A STABILITY DERIVATIVE RUN WILL BE EXECUTED
   60 CALL APPLY2
      IF(IR .EQ. 2) GO TO 120
C
C***********************************************************************
C   THIS RUN HAS BEEN COMPLETED.  THANK GOD FOR SMALL BLESSINGS.
C***********************************************************************
C
C   PRINT COMPLETION MESSAGE FOR THIS RUN AND GO BACK TO BEGIN A NEW RUN
   70 WRITE(6, 80 )
   80 FORMAT(1H0///, 32X,10(5H*****), 3H***/ 32X,
     1         53H*  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  */
     2         32X,10(5H*****),3H***)
C   READ TO SEE IF THE NEXT CARD IS A TITLE OR AN OLD END OF CMU CARD
      READ(LUN, 20, END=100) TITLE
   90 IF(TITLE(1) .EQ. CHECK) GO TO 10
      GO TO 30
C
C   PRINT COMPLETION MESSAGE AND STOP EXECUTION
  100 WRITE(6, 80 )
  110 STOP
C
C***********************************************************************
C   A FATAL ERROR HAS OCCURED.  PRINT FINAL MESSAGE AND STOP EXECUTION.
C***********************************************************************
  120 WRITE(6, 130 )
  130 FORMAT(1H0///62X,2(4H****)/31X,11(5H*****)/
     1    31X,55H*  THE  PROGRAM  HAS  REACHED  ABNORMAL  TERMINATION  */
     2    31X,11(5H*****)/62X,2(4H****))
  140 STOP
      END
C***********************************************************************
      SUBROUTINE CLRSCRN
C
C   LIBRARY ROUTINE TO CLEAR THE SCREEN.
C
      ISTAT = LIB$ERASE_PAGE (1,1)
      RETURN
      END
C***********************************************************************
      SUBROUTINE QUERY(NANS)
C
C   ROUTINE TO TRAP ERRORS CAUSED BY IMPROPER RESPONSES TO QUESTIONS.
C   THE COMPUTER GENERATES AND ERROR WHEN A CHARACTER IS SUPPLIED TO
C   A QUESTION EXPECTING AN INTEGER OR REAL VALUE.
C
      NQTEST=0
    1 CONTINUE
      IF (NQTEST .GT. 0) THEN
         PRINT *, '  CHARACTER VALUES ARE NOT VALID.'
         PRINT *, '  PLEASE ENTER AN INTEGER VALUE.'
      END IF
      NQTEST = NQTEST + 1
```

```
      READ (5,*,ERR=1)NANS
      RETURN
      END
C*****************************************************************
      SUBROUTINE APPLY1
C
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF REGULAR CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C  DECIDE WHETHER OR NOT THERE IS AN ALPHA CASE
      NOALFA = 1
      IR = 1
      LOGIC = 1
      IF(ISYMM .LT. 0) NOALFA = 0
C
C  INITIALIZE AND INCREMENT THE CMU CASE CONTROL COUNTER
   10 NEWCMU = 0
   20 NEWCMU = NEWCMU + 1
C
C  EXECUTE THE PROBLEM FORMATION STAGE
   30 CALL STAGE1
      GO TO ( 40 , 60 , 70 , 80 ), IR
C
C  EXECUTE THE PROBLEM SOLUTION STAGE
   40 CALL STAGE2
      IF(IR .EQ. 2) GO TO 80
C
C  EXECUTE THE AERODYNAMIC PARAMETER STAGE
   50 CALL STAGE3
C
C  THE PROGRAM HAS BEEN EXECUTED SUCESSFULLY
C  GO BACK AND DO A NEW CMU CASE
      IF(JETFLG .NE. 0) GO TO 60
      GO TO 20
C
C*****************************************************************
C  THIS RUN HAS BEEN COMPLETED.  RETURN TO START A NEW RUN.
   60 IR = 2
      RETURN
C  THIS RUN HAS BEEN COMPLETED.  NO FURTHER RUNS FOLLOW.
   70 IR = 3
      RETURN
C  A FATAL ERROR HAS OCCURED.  RETURN AND QUIT.
   80 IR = 4
      RETURN
C*****************************************************************
      END
      SUBROUTINE APPLY2
C
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF
C  STABILITY DERIVATIVES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C  CHECK ON STATUS OF CONTROL FLAGS
   10 IHINGE = 0
      NOALFA =1
      NEWCMU = 1
      IF(ISYMM .GE. 0) GO TO 30
      ISYMM = 0
      WRITE(6, 20 )
   20 FORMAT(1H0///16X,41HTHE ISYMM FLAG INDICATED AN ANTI-SYMETRIC,
     1     48H CASE.  HOWEVER, IT WILL BE TREATED AS SYMETRIC.)
C
C  EXCECUTE THE FIRST RUN
C
C  FORMULATE THE PROBLEM AS USUAL
   30 CALL STAGE1
      GO TO ( 40 , 110 , 100 , 110 ), IR
C
C  ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING
   40 LOGIC = 1
      CALL STAGE4
C
C  EXECUTE THE PROBLEM SOLUTION STAGE AS USUAL FOR THE FIRST RUN
      LOGIC = 1
   50 CALL STAGE2
      IF(IR .EQ. 2) GO TO 110
C
C  EXECUTE THE AERODYNAMIC STAGE FOR THE FIRST RUN
C  FUNDAMENTAL CASES
   60 LOGIC = 2
      CALL STAGE3
C
C  THE FIRST RUN HAS BEEN COMPLETED
C  EXECUTE THE SECOND RUN
C
      WRITE(6, 70  )
   70 FORMAT(1H1//////////// 37X,11(4H****),2H** /
     1          37X,46H*  SECOND RUN FOR STABILITY DERIVATIVE CASE  * /
     2          37X,11(4H****),2H**)
C  IF THIS IS A SYMETRIC WING, SWITCH IT TO ANTI-SYMETRIC FOR RUN 2
   80 IF(ISYMM .EQ. 0) ISYMM = -1
C
C  STORE THE FIRST RUN SOLUTION ON UNIT 1, DEFINE THE FUNDAMENTAL CASES
C  FOR YAWING AND ROLLING RATES, AND PRINT THE NEW FUND CASE GEOMETRY.
   90 LOGIC = 2
```

```
      CALL STAGE4
C
C  SET UP AND SOLVE THE MATRIX SYSTEM FOR THE SECOND RUN
      LOGIC = 2
      CALL STAGE2
      IF(IR .EQ. 2) GO TO 110
C
C  CALCULATE AND PRINT THE DERIVATIVES FOR ALL FUNDAMENTAL
C  AND COMPOSITE CASES
      LOGIC = 3
      IF(IPRINT .GE. 0) IPRINT = 2
      CALL STAGE3
C
C*****************************************************************
C  THIS IS THE END OF THE LINE
  100 IR = 1
      RETURN
C
C  A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.
  110 IR = 2
      RETURN
C*****************************************************************
      END
      SUBROUTINE STAGE1
C
C  THIS SUBROUTINE READS THE GENERAL GEOMETRY PARAMETERS AND FLAGS, AND
C  CONTROLS THE CALLING OF THE SPECIALIZED GEOMETRY SUBROUTINES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C  CHECK WHETHER THIS IS THE FIRST CMU CASE
      IF(NEWCMU .GT. 1) GO TO 50
      IF((NROWS .GT. 40) .OR. (NROWS .LT. 3)) GO TO 80
C
C  SECTIONAL INPUT
   10 IF((IGTYPE .EQ. 1).OR.(IGTYPE .EQ. 2))CALL SGMAIN(NOALFA,IR)
      GO TO ( 20 , 40 , 100 ), IR
C
C  USER INPUT
C
C  PRINT ERROR MESSAGE BECAUSE IGTYPE HAS THE WRONG VALUE
   20 WRITE(6, 30 ) IGTYPE
   30 FORMAT(1H1//44X,32HTHE IGTYPE FLAG HAS THE VALUE OF,I2/
     1         44X,37HONLY THE VALUES 1 OR 2 ARE ACCEPTABLE//
     2         44X,29HTHIS CASE HAS BEEN TERMINATED)
      GO TO 100
C
C  READ THE COMPOSITE CASE REQUIREMENTS
   40 CALL INCOMP(NCASES,IR)
      IF(IR .EQ. 2) GO TO 100
C
C  READ THE CMU DATA
   50 CALL BLOWIN(JETFLG,IR)
      GO TO ( 60 , 110 , 120 ), IR
   60 CALL BOXJ(NEWMAX,IR)
      IF(IR .EQ. 2) GO TO 50
C
C  RETURN NORMALLY TO THE CONTROL PROGRAM
   70 IR = 1
      GO TO 130
C
C  PRINT ERROR MESSAGE BECAUSE THE NROWS VALUE IS UNACCEPTABLE
   80 WRITE(6, 90 ) NROWS
   90 FORMAT(1H1/55X,7HNROWS =,I3)
C
C*****************************************************************
C  A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.
  100 IR = 4
      GO TO 130
C
C  THIS RUN HAS BEEN COMPLETED.   THANK GOD FOR SMALL BLESSINGS.
C
C  RETURN TO MAIN AND BEGIN A COMPLETELY NEW RUN
  110 IR = 2
      GO TO 130
C
C  RETURN TO MAIN AND STOP THE EXECUTION
  120 IR = 3
  130 RETURN
C*****************************************************************
      END
      SUBROUTINE SGMAIN(NOALFA,IR)
C
C  THIS SUBROUTINE CONTROLS ALL GEOMETRY CALCULATIONS FOR THE
C  SECTIONAL GEOMETRY METHOD
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
C
C  READ THE WING PLANFORM GEOMETRY DATA
   10 CALL INPTS(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 1) CALL XLETR1(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 2) CALL XLETR2
C
C  NORMALIZE THE WING PLANFORM GEOMETRY DATA
   20 CALL NORM1
C
C  READ THE JET SHEET GEOMETRY DATA
```

```
   30 CALL INPUTJ(IR)
      IF(IR .EQ. 2) GO TO 100
C
C  CONSTRUCT THE EVD ELEMENTS
   40 CALL BOXS(IR)
      IF(IR .EQ. 2) GO TO 100
C
C  CONSTRUCT THE SET OF FUNDAMENTAL GEOMETRIC CASES
C
      DO 90 N = 1,NCASES
      LCASE = N
C
C  READ THE GEOMETRY FOR THIS CASE
   50 CALL INCASE(LCASE,NOALFA)
C
C  CONSTRUCT THE CASE DATA
   60 CALL BEECEE(LCASE,NOALFA,IR)
      IF(IR .EQ. 2) GO TO 100
C
C  PRINT THE GEOMETRY AND CONSTRUCTED CASE DATA IF REQUIRED
      IF(LCASE .EQ. 1) WRITE(6, 70 )
   70 FORMAT(1H1)
   80 CALL OUT1(LCASE)
   90 CONTINUE
      IR = 2
      RETURN
C
C  AN ERROR HAS OCCURED.   RETURN ABNORMALLY TO STAGE1.
C
  100 IR = 3
      RETURN
      END
      SUBROUTINE INPTS(IR)
C
C  THIS SUBROUTINE READS THE WING GEOMETRY DATA
C  FOR THE SECTIONAL GEOMETRY METHOD
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1             NWTYPE,NJTYPE
      COMMON/INDAT/LUN
      DIMENSION NI(10)
C
C  READ THE SECTIONAL PLANFORM DATA
   10 NWTYPE = 0
      READ(LUN, 20 ) (Y(K),K=1,NROWS)
   20 FORMAT(8F10.6)
      READ(LUN, 30 ) (ICTYPE(K),K=1,NROWS)
   30 FORMAT(40I2)
      DO 40 K = 1,NROWS
      IF(ICTYPE(K) .GT. NWTYPE) NWTYPE = ICTYPE(K)
   40 CONTINUE
      IF(NWTYPE .GT. 10) GO TO 80
      READ(LUN, 30 )(NI(N),N=1,NWTYPE)
C
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
      DO 50 N = 1,NWTYPE
      NIN = NI(N)
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 100
      READ(LUN, 20 ) (XBW(L,N),L=1,NIN)
   50 CONTINUE
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW
      DO 70  K = 1,NROWS
      ICK = ICTYPE(K)
   60 NW(K) = NI(ICK)
   70 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.
   80 WRITE(6, 90 ) NWTYPE
   90 FORMAT(1H1/45X,26HNUMBER OF WING ROW TYPES =,I3)
      IR = 2
      RETURN
  100 WRITE(6, 110 ) NIN,N
  110 FORMAT(1H1/38X,I3,38H WING ELEMENTS PRESCRIBED FOR ROW TYPE,I3)
      IR = 2
      RETURN
      END
      SUBROUTINE INPUTJ(IR)
C
C  THIS SUBROUTINE READS THE JET GEOMETRY INPUT
C  FOR THE SECTIONAL GEOMETRY METHOD
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1             NWTYPE,NJTYPE
      COMMON/INDAT/LUN
      DIMENSION NI(10)
C
C  READ THE TYPE OF DIVISION FOR EACH ROW
   10 NJTYPE = 0
      NROWSJ = 0
      IF(JETFLG .NE. 0) GO TO 90
      READ(LUN, 20 ) (IJTYPE(K),K=1,NROWS)
   20 FORMAT(40I2)
      DO 30 K = 1,NROWS
```

```
      IF(IJTYPE(K) .GT. NJTYPE) NJTYPE = IJTYPE(K)
      IF(IJTYPE(K) .NE. 0) NROWSJ = NROWSJ + 1
   30 CONTINUE
      IF(NJTYPE .GT. 10) GO TO 110
C  READ THE NUMBER OF CHORDWISE DIVISIONS IN EACH ROW TYPE
      READ(LUN, 40 ) (NI(N),N=1,NJTYPE)
   40 FORMAT(10I2)
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
      DO 60 N = 1,NJTYPE
      NIN = NI(N)
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 130
      READ(LUN, 50 ) (XBJ(L,N),L=1,NIN)
   50 FORMAT(8F10.6)
   60 CONTINUE
C
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW
      DO 80  K = 1,NROWS
      NJ(K) = 0
      IF(IJTYPE(K) .EQ. 0) GO TO 80
      IJK = IJTYPE(K)
   70 NJ(K) = NI(IJK)
   80 CONTINUE
C  CHECK FOR ROW CONSISTENCY ON EITHER SIDE OF JET
      ICOUNT = 1
      IF(NJ(1) .EQ. 0)ITEST = 0
      IF(NJ(1) .GT. 0)ITEST = 1
      DO 150 K = 2,NROWS
      IF(NJ(K) .EQ. 0)ICOMP = 0
      IF(NJ(K) .GT. 0)ICOMP = 1
      IF(ICOMP .EQ. ITEST)GO TO 160
      IF(ICOUNT .LT. 3)GO TO 170
      ICOUNT = 1
      IF(NJ(K) .EQ. 0)ITEST = 0
      IF(NJ(K) .GT. 0)ITEST = 1
      GO TO 150
  160 ICOUNT = ICOUNT + 1
  150 CONTINUE
      IF(ICOUNT .LT. 3) GO TO 170
      IR = 1
      RETURN
C
C  THERE IS NO JET FOR THIS RUN
   90 DO 100 K = 1,NROWS
      IJTYPE(K) = 0
      NJ(K) = 0
  100 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.    PRINT A MESSAGE AND QUIT.
  110 WRITE(6, 120 ) NJTYPE
  120 FORMAT(1H1/25HNUMBER OF JET ROW TYPES =,I3)
      IR = 2
      RETURN
  130 WRITE(6, 140 ) NIN,N
  140 FORMAT(1H1/38X,I3,37H JET ELEMENTS PRESCRIBED FOR ROW TYPE,I3)
      IR = 2
      RETURN
  170 WRITE(6,190)
  190 FORMAT(1H1,29H3 ROW CONTINUITY RULE FAILURE)
      IR = 2
      RETURN
      END
      SUBROUTINE XLETR1(IR)
C
C  THIS SUBROUTINE READS THE LEADING AND TRAILING EDGE COORDINATES
C  AT SPANWISE STATIONS CONNECTED BY STRAIGHT LEADING AND TRAILING EDGES
C  AND INTERPOLATES TO GET THE COORDINATES AT INTERMEDIATE SECTIONS
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/INDAT/LUN
      DIMENSION YP(40),XLE(40),XTR(40)
C
C  READ XLEAD AND XTRAIL
      NX = 0
      DO 30 N = 1,NROWS
      READ(LUN, 10 )YP(N),XLE(N),XTR(N)
   10 FORMAT(3F10.6)
   20 IF(YP(N) .GT. 1.1) GO TO 40
      NX = NX + 1
   30 CONTINUE
C  CHECK WHETHER THE Y VALUES ARE REALISTIC
   40 IF(ABS(YP(1)-Y(1)) .GT. 0.0001)GO TO 110
      IF(ABS(YP(NX)-Y(NROWS)) .GT. 0.0001)GO TO 110
C
C  READ THE EXTRA 9 CARD IF NROWS CARDS HAVE BEEN INPUT
      IF(NX .EQ. NROWS) READ(LUN, 10 ) EXTRA9
C
C  INTERPOLATE FOR XLEAD AND XTRAIL AT THE INTERMEDIATE SECTIONS
      K = 0
      DO 100 N = 1,NX
   50 K = K + 1
      IF(K .GT. NROWS) GO TO 110
      IF(ABS(YP(N)-Y(K)) .GT. 0.0001)GO TO 70
C  XLE AND XTR WERE INPUT FOR ROW K
   60 XLEAD(K) = XLE(N)
      XTRAIL(K) = XTR(N)
```

```
      GO TO 100
C  XLE AND XTR MUST BE INTERPOLATED FOR ROW K
   70 NM1 = N - 1
   80 YRATIO  = (Y(K)-YP(N)) / (YP(NM1)-YP(N))
      XLEAD(K) = XLE(N) + YRATIO * (XLE(NM1)-XLE(N))
   90 XTRAIL(K) = XTR(N) + YRATIO * (XTR(NM1)-XTR(N))
      GO TO 50
  100 CONTINUE
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND RETURN.
  110 WRITE(6, 120 )
  120 FORMAT(1H1//20X,38HAN INCONSISTENCY HAS BEEN FOUND IN THE,
     1       42H SECTIONAL LEADING AND TRAILING EDGE INPUT)
      IR = 2
      RETURN
      END
      SUBROUTINE XLETR2
C
C  THIS SUBROUTINE READS THE FUNDAMENTAL PLANFORM PARAMETERS FOR A
C  TRAPEZOIDAL WING, AND CALCULATES THE LEADING AND TRAILING EDGE
C  COORDINATES AT EACH Y STATION.  NOTE THAT THE PLANFORM OUTLINE
C  MUST BE SYMETRIC.
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/INDAT/LUN
C
C  READ THE FUNDAMENTAL PLANFORM PARAMETERS
      READ(LUN, 10 ) ARATIO,SWEEP,TR
   10 FORMAT(4F10.6)
C
C  COMPUTE THE GENERAL PLANFORM CHARACTERISTICS
      B2 = SPAN / 2.00
      SW = SWEEP / 57.295779
   20 CROOT = 2.0 * SPAN / ((1.0+TR)*ARATIO)
      AREA = (1.0+TR) * CROOT * B2
      XLB2 = 0.250 * (1.0-TR) * CROOT + B2 * TAN(SW)
   30 CMAC = 2.0 * CROOT * (1.0 + TR + TR*TR) / (3.0*(1.0+TR))
      IF(CREF .EQ. 0.0) CREF = CMAC
      CBAR=AREA/SPAN
C
C  COMPUTE THE LEADING AND TRAILING EDGE COORDINATES
   40 DO 60 K = 1,NROWS
      YBAR = Y(K)
      IF(YBAR .LT. 0.0) YBAR = -YBAR
      XLEAD(K) = XLB2 * YBAR
   50 C = CROOT * (1.0-(1.0-TR)*YBAR)
      XTRAIL(K) = XLEAD(K) + C
   60 CONTINUE
      RETURN
      END
      SUBROUTINE NORM1
C
C  THIS SUBROUTINE NORMALIZES ALL WING PLANFORM GEOMETRY BY SPAN/2
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
   10 B2 = SPAN / 2.00
      AREA = AREA / B2**2
      CREF = CREF / B2
   20 XMC = XMC / B2
      XCG = XCG / B2
      DO 40 K = 1,NROWS
   30 XLEAD(K) = XLEAD(K) / B2
      XTRAIL(K) = XTRAIL(K) / B2
   40 CONTINUE
      SPAN = 2.00
      ARATIO = SPAN * SPAN / AREA
      RETURN
      END
      SUBROUTINE BOXS(IR)
C
C  THIS SUBROUTINE CONSTRUCTS THE GEOMETRIC PARAMETERS FOR ALL THE
C  EVD ELEMENTS ON THE WING AND JET
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1             NWTYPE,NJTYPE
C
C  CONSTRUCT THE ELEMENTS ON THE WING
C
C  COMPUTE SECTIONAL DATA
   10 CHORD(1) = XTRAIL(1) - XLEAD(1)
      DELTA(1) = 1.00 - Y(1)
      CMAC = CHORD(1)**2 * DELTA(1)
      DO 30 K = 2,NROWS
   20 CHORD(K) = XTRAIL(K) - XLEAD(K)
      DELTA(K) = Y(K-1) - Y(K) - DELTA(K-1)
```

```
      IF(DELTA(K) .LT. 0.0) GO TO 190
      CMAC = CMAC + CHORD(K)**2 * DELTA(K)
   30 CONTINUE
C  CHECK THE VALIDITY OF THE SECTIONAL ALLIGNMENT
      YD = Y(NROWS) - DELTA(NROWS)
      IF((ISYMM .GE. 0) .AND. (ABS(YD) .GT. 0.0001)) GO TO 190
      IF((ISYMM .EQ. 1) .AND. (ABS(YD+1.0) .GT. 0.0001)) GO TO 190
      DSUM = DELTA(1)
      DO 35 K = 2,NROWS
      YL = Y(K) + DELTA(K)
      YR = Y(K-1) - DELTA(K-1)
      IF(ABS(YR-YL) .GT. 0.0001) GO TO 190
      DSUM = DSUM + DELTA(K)
   35 CONTINUE
      IF(ABS(DSUM-0.50) .GT. 0.0001) GO TO 190
      CMAC = 2.0 * CMAC / AREA
      IF(ISYMM .LT. 1) CMAC = 2.0 * CMAC
      IF(CREF .LT. 0.0001) CREF = CMAC
      CALL TANS(TANLE,XLEAD,Y,NROWS)
C
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      I = 0
      DO 90  K = 1,NROWS
C  COMPUTE X-COORDINATES
      NWK = NW(K)
      DO 50 L = 1,NWK
      I = I + 1
      ICK = ICTYPE(K)
   40 XB(I) = XBW(L,ICK)
   50 CONTINUE
C  COMPUTE ALL OTHER PARAMETERS
      I = I - NWK
      IW(K) = I + 1
      DO 80 L = 1,NWK
      I = I + 1
   60 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
   70 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
   80 CONTINUE
C  REDEFINE THE LAST DEL IN THIS SECTION, AND DEFINE THE L.E. EVD TYPE
      DEL(I) = 1.00 - XB(I)
      IWK = IW(K)
      ITYPE(IWK) = 20
   90 CONTINUE
      NWT = I
C
C  CONSTRUCT THE ELEMENTS ON THE JET SHEET
C
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      IF(JETFLG .NE. 0) GO TO 180
      DO 170 K = 1,NROWS
C  COMPUTE X-COORDINATES
      IJ(K) = 0
  100 NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 170
      DO 120 L = 1,NJK
      I = I + 1
      IJK = IJTYPE(K)
  110 XB(I) = XBJ(L,IJK)
  120 CONTINUE
C  COMPUTE ALL OTHER PARAMETERS
      I = I - NJK
  130 IJ(K) = I + 1
      DO 160 L = 1,NJK
      I = I + 1
  140 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
  150 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
  160 CONTINUE
C  REDEFINE THE LAST DEL AND EVD TYPE AND THE D VALUE FOR THIS SECTION
      DEL(I) = 1.0E10
      ITYPE(I) = 30
      D(K) = XI(I) - XTRAIL(K)
  170 CONTINUE
  180 NMAX = I
      IF(NMAX .GT. 600) GO TO 210
      NJT = NMAX - NWT
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.
  190 WRITE(6, 200 )
  200 FORMAT(1H1/38X,44HPLEASE CHECK YOUR SECTION LOCATION (Y) INPUT)
      IR = 2
      RETURN

  210 WRITE(6, 220 ) NMAX
  220 FORMAT(1H1/48X,I4,21H IS TOO MANY ELEMENTS)
      IR = 2
      RETURN
      END
      SUBROUTINE BOXJ(NEWMAX,IR)
C
C  THIS SUBROUTINE COMPUTES THE JET BLOWING FACTOR CMUP
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
```

```
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
C
C  COMPUTE THE NEW CMUP AND SAVE THE OLD VALUES AS CMUPP
   10 NEWMAX = NMAX
      ICOUNT = 0
      DO 30 K = 1,NROWS
      CMUPP(K) = CMUP(K)
      IF(NJ(K) .EQ. 0) GO TO 30
      IF(CMU(K) .LT. 0.0001) GO TO 20
      CMUP(K) = 2.00 / (CHORD(K)*CMU(K))
      GO TO 30
   20 ICOUNT = ICOUNT + 1
      CMUP(K) = 0.00
   30 CONTINUE
      WRITE(6, 40 ) (K,CMU(K),K=1,NROWS)
   40 FORMAT(1H1,40X,10(4H****)/ 41X,
     1       40H*   SECTIONAL JET BLOWING COEFFICIENTS   */41X,10(4H****)//
     2       53X,3HROW,5X,3HCMU,40(/53X,I2,F12.6))
      IF(ICOUNT .EQ. 0) GO TO 50
      IF(ICOUNT .LT. NROWSJ) GO TO 60
      NEWMAX = NWT
   50 IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND TRY AGAIN.
   60 WRITE(6, 70 )
   70 FORMAT(1H0,43X,35HA ZERO VALUE OF CMU HAS BEEN INPUT.,
     1       35H   THIS CMU CASE HAS BEEN IGNORED.)
      IR = 2
      RETURN
      END
      SUBROUTINE TANS(TAN,X,Y,NROWS)
C
C  THIS SUBROUTINE COMPUTES THE TANGENT OF THE LEADING OR TRAILING EGDE
C  SWEEP ANGLE AT THE CENTERLINE OF EACH SECTION.   IT IS ACCURATE FOR
C  SECTIONS WITH STRAIGHT EDGES IN GROUPS OF THREE OR MORE.
C  IT IS ONLY APPROXIMATE FOR CURVED EDGES.
C  IT MAY RESULT IN ERRORS FOR SECTIONS ADJACENT TO WING BREAKS,
C  IF STRAIGHT EDGES ARE IN ADJACENT GROUPS OF ONLY ONE OR TWO.
C
      DIMENSION TAN(40),X(40),Y(40),S(40)
      SLOP(XR,XL,YR,YL) = (XR-XL) / (YR-YL)
C
      DO 50 K = 1,NROWS
      KR = K-1
      KL = K
      IF(K .GT. 1) GO TO 30
      KR = 1
      KL = 2
   30 S(K) = SLOP(X(KR),X(KL),Y(KR),Y(KL))
   50 CONTINUE
      DO 200 K = 1,NROWS
      IF(K .LT. 3) GO TO 150
      IF(K .EQ. NROWS) GO TO 150
      IF(K .EQ. (NROWS-1)) GO TO 160
C  CHECK WHETHER THE RIGHT OR LEFT SIDES ARE STRAIGHT
      IF(ABS(S(K) - S(K-1)) .LT. 0.001) GO TO 150
      IF(ABS(S(K+1) - S(K+2)) .LT. 0.001) GO TO 160
C  NEITHER SIDE IS CONCLUSIVELY STRAIGHT - CHECK FURTHER LEFT AND RIGHT
      IF(K .EQ. 3) GO TO 160
      IF(K .EQ. (NROWS-2)) GO TO 150
      IF(ABS(S(K-1) - S(K-2)) .LT. 0.001) GO TO 160
      IF(ABS(S(K+2) - S(K+3)) .LT. 0.001) GO TO 150
C  THE TRUE SHAPE CANNOT BE DETERMINED - GIVE UP AND TAKE THE AVERAGE
      TAN(K) = (S(K) + S(K+1)) / 2.00
      GO TO 200
C  THE RIGHT EDGE IS STRAIGHT
  150 TAN(K) = S(K)
      GO TO 200
C  THE LEFT EDGE IS STRAIGHT
  160 TAN(K) = S(K+1)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE INCASE(LCASE,NOALFA)
C
C  THIS SUBROUTINE READS THE FUNDAMENTAL GEOMETRIC CASE DATA
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTB
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/INDAT/LUN
      DIMENSION NI(10),DUMMY(40)
C
C  INITIALIZE SECTIONAL DATA
      DO 30 K = 1,NROWS
   10 TST(K,LCASE) = 0.00
      HL(K,LCASE) = 0.00
      DJ(K) = 0.00
      ACTE(K) = 0.00
      ICT(K) = 0
      IHT(K) = 0
C  INITIALIZE THE CAMBER ANGLES
      NWK = NW(K)
      DO 20 L = 1,NWK
      AC(L,K) = 0.00
   20 CONTINUE
   30 CONTINUE
C  INITIALIZE THE HINGE DATA
      DO 50 N = 1,NROWS
```

```
      DO 40 L = 1,4
      XHB(L,N) = 0.00
      BET(L,N) = 0.00
   40 CONTINUE
   50 CONTINUE
C
      IF((LCASE .EQ. 1) .AND. (NOALFA .GT. 0)) RETURN
C
C   READ FUNDAMENTAL CASE CONTROL FLAGS
      READ(LUN, 60 ) INPUTT,INPUTH,INPUTD,INPUTC,INPUTB
   60 FORMAT(5I2)
C
C   READ SECTIONAL TWIST, HEIGHT AND JET DEFLECTION DATA
      IF(INPUTT .NE. 0) READ(LUN, 70 ) (TST(K,LCASE),K=1,NROWS)
   70 FORMAT(8F10.6)
      IF(INPUTH .NE. 0) READ(LUN, 70 ) (HL(K,LCASE),K=1,NROWS)
      IF(INPUTD .EQ. 0) GO TO 90
      READ(LUN, 70 ) (DUMMY(K),K=1,NROWSJ)
C   DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE DJ ARRAY
      KP = 0
      DO 80 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 80
      KP = KP + 1
      DJ(K) = DUMMY(KP)
   80 CONTINUE
C
C   READ THE CAMBER ANGLES, IN DEGREES
   90 IF(INPUTC .EQ. 0) GO TO 160
      READ(LUN, 100 ) (ICT(K),K=1,NROWS)
  100 FORMAT(40I2)
      NCT = 0
      DO 110 K = 1,NROWS
      IF(ICT(K) .EQ. 0) GO TO 110
      IF(ICT(K) .GT. NCT) NCT = ICT(K)
      ICK = ICT(K)
      NI(ICK) = NW(K)
  110 CONTINUE
      DO 130 N = 1,NCT
      NIN = NI(N)
  120 READ(LUN, 70 ) (AC(L,N),L=1,NIN)
  130 CONTINUE
  140 IF(NROWSJ .GT. 0) READ(LUN, 70 ) (DUMMY(K),K=1,NROWSJ)
C   DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE ACTE ARRAY
      KP = 0
      DO 150 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 150
      KP = KP + 1
      ACTE(K) = DUMMY(KP)
  150 CONTINUE
C
C   READ THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA
  160 IF(INPUTB .EQ. 0) GO TO 210
  170 READ(LUN, 100 ) (IHT(K),K=1,NROWS)
      NHT = 0
      DO 180 K = 1,NROWS
      IF(IHT(K) .GT. NHT) NHT = IHT(K)
  180 CONTINUE
      DO 200 N = 1,NHT
      READ(LUN, 190 ) (XHB(L,N),IFS(L,N),BET(L,N),L=1,4)
  190 FORMAT(4(F10.6,I1,F9.6))
  200 CONTINUE
  210 RETURN
      END
      SUBROUTINE BEECEE(LCASE,NOALFA,IR)
C
C   THIS SUBROUTINE CONSTRUCTS THE BOUNDARY CONDITION ARRAYS FOR THE
C   FUNDAMENTAL GEOMETRIC CASES
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTB
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION NH(40)
C
C   INITIALIZE THE BOUNDARY CONDITION ANGLES
      DO 20 I = 1,NMAX
   10 EPS(I,LCASE) = 0.00
      BETA(I,LCASE) = 0.00
   20 CONTINUE
      DO 30 K = 1,NROWS
      THETA(K,LCASE) = 0.00
      THS(K,LCASE) = 0.00
   30 CONTINUE
C
C   DEFINE THE ANGLES FOR THE ANGLE-OF-ATTACK FUNDAMENTAL CASE
      IF((LCASE .GT. 1) .OR. (NOALFA .EQ. 0)) GO TO 60
      DO 40 I = 1,NWT
      EPS(I,1) = 1.00
   40 CONTINUE
      DO 50 K = 1,NROWS
      IF(NJ(K) .GT. 0) THETA(K,1) = 1.000
   50 CONTINUE
      IR = 1
      RETURN
C
C   DEFINE THE ANGLES FOR ALL REMAINING FUNDAMENTAL CASES
C
C   CAMBER CONTRIBUTION
```

```
   60 IF(INPUTC .EQ. 0) GO TO 110
      I = 0
      DO 100 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 70
      THETA(K,LCASE) = ACTE(K)
   70 NHK = NH(K)
      IF(ICT(K) .EQ. 0) GO TO 90
      DO 80 L = 1,NHK
      I = I + 1
      ICK = ICT(K)
      EPS(I,LCASE) = AC(L,ICK)
   80 CONTINUE
      GO TO 100
   90 I = I + NHK
  100 CONTINUE
C
C  TWIST CONTRIBUTION
  110 IF(INPUTT .EQ. 0) GO TO 160
      I = 0
      DO 150 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 120
      THETA(K,LCASE) = THETA(K,LCASE) + TST(K,LCASE)
  120 NHK = NH(K)
      DO 140 L = 1,NHK
      I = I + 1
  130 EPS(I,LCASE) = EPS(I,LCASE) + TST(K,LCASE)
  140 CONTINUE
  150 CONTINUE
C
C  FLAP AND SLAT DEFLECTION CONTRIBUTION
C
  160 IF(INPUTB .EQ. 0) GO TO 320
C
C  SUM UP THE TOTAL SLAT ANGLE AND FIND THE NUMBER OF HINGES ON EACH ROW
      DO 190 K = 1,NROWS
      NH(K) = 0
      IF(IHT(K) .EQ. 0) GO TO 190
      DO 180 L = 1,4
  170 N = IHT(K)
      IF(XHB(L,N) .LT. 0.001) GO TO 180
      NH(K) = NH(K) + 1
      IF(IFS(L,N) .GT. 0) THS(K,LCASE) = THS(K,LCASE) + BET(L,N)
  180 CONTINUE
  190 CONTINUE
C
C  COMPUTE INCIDENCE OF EACH ELEMENT AND FIND TURNING ANGLE AND EVD TYPE
C  FOR EACH HINGE ELEMENT
      I = 0
      DO 310 K = 1,NROWS
      NNK = NN(K)
      N = IHT(K)
  200 IF(N .EQ. 0) GO TO 300
      LSTART = 1
      B = 0.0
      NHK = NH(K)
  210 IF(NHK .EQ. 0) GO TO 360
C  CYCLE THE HINGE POINTS IN CHORDWISE ORDER
      DO 270 LH = 1,NHK
C  CYCLE THE VORTEX POINTS IN CHORDWISE ORDER, LOOKING FOR NEXT HINGE
      DO 250 L = LSTART,NNK
      I = I + 1
C  CHECK ON RELATIVE LOCATION OF VORTEX POINT AND HINGE POINT
  220 XDIFF = XHB(LH,N) - XB(I)
      IF(ABS(XDIFF) .LT. 0.001) GO TO 230
      IF(XDIFF .GT. 0.001) GO TO 240
      IF(XDIFF .LT. -0.001) GO TO 360
C  THE ITH VORTEX POINT IS A HINGE POINT
  230 B = B + BET(LH,N)
      BETA(I,LCASE) = BET(LH,N)
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
      ITYPE(I) = 42
      IF(IFS(LH,N) .GT. 0) ITYPE(I) = 41
      GO TO 260
C  THE ITH VORTEX POINT IS NOT A HINGE POINT
  240 EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
  250 CONTINUE
  260 LSTART = I - IN(K) + 1
  270 CONTINUE
C  DEFINE THE INCIDENCE ANGLE FOR REMAINING POINTS BEHIND THE LAST HINGE
      IF(LSTART .EQ. NNK) GO TO 290
      LSTART = LSTART + 1
      DO 280 L = LSTART,NNK
      I = I + 1
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B
  280 CONTINUE
C  COMPUTE THE EFFECT OF THE HINGES ON THE JET ANGLE
  290 IF(NJ(K) .GT.  0) THETA(K,LCASE) = THETA(K,LCASE)-THS(K,LCASE)+B
      GO TO 310
  300 I = I + NNK
  310 CONTINUE
C
C  JET DEFLECTION CONTRIBUTION
C
  320 IF(INPUTD .EQ. 0) GO TO 350
      DO 340 K = 1,NROWS
      IF(NJ(K) .EQ. 0) GO TO 340
      I = IJ(K)
  330 BETA(I,LCASE) = DJ(K)
      IF(ABS(DJ(K)) .GT. 0.0001) ITYPE(I) = 43
      THETA(K,LCASE) = THETA(K,LCASE) + DJ(K)
  340 CONTINUE
```

```
  350 IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.
C
  360 WRITE(6, 370 ) LCASE,K,N
  370 FORMAT(1H1//45X,26HFUNDAMENTAL GEOMETRIC CASE,I3/
     1        18X,50HAN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT,
     2        18H DATA FOR WING ROW,I3,10H, ROW TYPE,I3)
      IR = 2
      RETURN
      END
      SUBROUTINE OUT1(LCASE)
C
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE
C  SECTIONAL METHOD INPUT
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(20)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
C
C  PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS
      IF(LCASE .GT. 1) GO TO 60
   10 WRITE(6, 20 )TITLE
   20 FORMAT(1H1,39X,10(4H****)/
     1    40X,40H*  EVD  JET - WING  COMPUTER  PROGRAM  */
     2    40X,10(4H****)///20X,20A4)
      CMA = CMAC * SPA / 2.0
   30 WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,
     1             ARATIO,XCG,XC
   40 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /
     1        41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /
     2        41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /
     3        41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /
     4        42X,5HXCG =,2F15.6)
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,
     1             IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX
   50 FORMAT(1H0/ 48X,7HNROWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /
     1             48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /
     2             47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /
     3             47X,8HIHINGE =,I3,7X,I3 ///
     4             43X,25HNUMBER OF WING ELEMENTS =,I4 /
     5             43X,25HNUMBER OF JET ELEMENTS  =,I4 /
     6             42X,26HTOTAL NUMBER OF ELEMENTS =,I4)
   60 J = 0
      JJ = NWT
C
C  PRINT FUNDAMENTAL CASE HEADER
      WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,23X,1H*,19(4H****)/
     1    24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,
     2    17H FUNDAMENTAL CASE,I3,3H  */24X,1H*,19(4H****))
      ILINES = 3
      DO 260 K = 1,NROWS
C
C  PRINT SECTIONAL DATA
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)
   80 FORMAT(1H0,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,
     2 2X,7HTANLE =,F10.6)
C
C  PRINT CHORDWISE DATA ON WING
      NWK = NW(K)
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)
   90 FORMAT(1H0,21HWING ELEMENTS    NW =,I3,5X, 7HTWIST =,F10.6,5X,
     1             4HHL =,F10.6,5X, 9HTHETA S =,F10.6)
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)
  100 FORMAT(1H ,14X,2HXB,10F11.6 / 17X,10F11.6)
      IF(LCASE .GT. 1) GO TO 130
      WRITE(6, 110 ) (XI(J+L),L=1,NWK)
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)
  130 IF(ICT(K) .EQ. 0) GO TO 150
      ICK = ICT(K)
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)
  140 FORMAT(1H ,10X,6HCAMBER,10F11.6/17X,10F11.6)
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))
      J = J + NWK
      IL = 1
      IF(NWK .GT. 9) IL = 2
      ILINES = ILINES + 4 + 4*IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
C
C  PRINT CHORDWISE DATA ON JET
      NJK = NJ(K)
      IF(NJK .GT. 0) GO TO 200
      WRITE(6, 190 )
```

```
190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)
    ILINES = ILINES + 1
    GO TO 230
200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)
210 FORMAT(1H0,1X,20HJET ELEMENTS     NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,
   1     F10.6,5X,6HACTE =,F10.6,5X,7HTHETA =,F10.6)
    WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)
    IF(LCASE .GT. 1) GO TO 220
    WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)
    WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)
220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)
    WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)
    JJ = JJ + NJK
    IL = 1
    IF(NJK .EQ. 10) IL = 2
    ILINES = ILINES + 1 + 3 * IL
    IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
230 IF(K .EQ. NROWS) GO TO 260
    NWK1 = NW(K+1)
    IL = 1
    IF(NWK1 .GT. 9) IL = 2
    NEXT = 4 + 4*IL
    IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
    NJK1 = NJ(K+1)
    IL = 1
    IF(NJK1 .EQ. 10) IL = 2
    NEXT = NEXT + 1
    IF(NJK1 .EQ. 0) GO TO 240
    NEXT = NEXT + 1 + 3*IL
    IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
240 IF((55-ILINES) .GE. NEXT) GO TO 260
    WRITE(6, 250 )
250 FORMAT(1H1)
    ILINES = 1
260 CONTINUE
    RETURN
    END
    SUBROUTINE INCOMP(NCASES,IR)
C
C  THIS SUBROUTINE READS IN THE COMPOSITE CASE REQUIREMENTS
C  WHICH DEFINE THE FUNDAMENTAL CASES AND THEIR DEFLECTION MAGNITUDE
C  FOR SUPERPOSITION IN UP TO 24 COMBINATIONS
C
    COMMON/COMPOS/FACTOR(10,24),NCC
    COMMON/INDAT/LUN
    DIMENSION FUNNY(10),ND(10)
C
C  INITIALIZE THE ARRAY OF FUNDAMENTAL CASE DEFLECTIONS
    DO 20 M = 1,24
    DO 10 N = 1,10
    FACTOR(N,M) = 0.00
 10 CONTINUE
 20 CONTINUE
C
C  READ THE COMPOSITE CASE DATA, CONSISTING OF FUNDAMENTAL CASE
C  DEFLECTIONS, IN DEGREES
    NCC = 0
 30 NCC = NCC + 1
    READ(LUN, 40, END=130 ) (ND(N),FUNNY(N),N=1,10)
 40 FORMAT(10(BZ,I2,F6.4))
C  CHECK THE VALIDITY OF THE DATA
 50 IF(ND(1) .GT. 10) GO TO 100
    IF(NCC .GT. 24) GO TO 110
    DO 90 N = 1,10
    IF(ND(N) .GT. NCASES) GO TO 70
    IF(ND(N) .LT. 1) GO TO 90
C  THE DATA IS OK.  DEFINE FACTOR.
    NDN = ND(N)
 60 FACTOR(NDN,NCC) = FUNNY(N)
    GO TO 90
 70 WRITE(6, 80 )
 80 FORMAT(1H0,22X,76HAN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN
   1 FOUND.  IT WILL BE IGNORED.)
 90 CONTINUE
    GO TO 30
C  THE END OF THE INPUT DATA HAS BEEN REACHED
100 NCC = NCC - 1
    IF(NCC .GT. 24) NCC = 24
    IR = 1
    RETURN
C  TOO MANY COMPOSITE CASES HAVE BEEN REQUESTED.  READ ON UNTIL AN END
C  CARD IS FOUND.
110 WRITE(6, 120 )
120 FORMAT(1H0,20X,47HMORE THAN 24 COMPOSITE CASES HAVE BEEN INPUT.  ,
   1          34HSUBSEQUENT INPUTS WILL BE IGNORED.)
    GO TO 30
C
C  AN END OF FILE HAS BEEN READ.  PRINT A MESSAGE AND QUIT.
130 WRITE(6, 140 )
140 FORMAT(1H1/// 31X, 35HAN END OF FILE HAS BEEN READ DURING,
   1       21H COMPOSITE CASE INPUT)
    IR = 2
    RETURN
    END
    SUBROUTINE BLOWIN(JETFLG,IR)
C
C  THIS SUBROUTINE READS THE SECTIONAL JET BLOWING RATES
C  CMU(K) = J / (Q * CHORD(K))
C
    COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
    COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
```

```
      DIMENSION DCMU(40)
      COMMON/INDAT/LUN
C
      IF(JETFLG .NE. 0) GO TO 30
C  READ THE CMU DATA ONLY FOR THOSE SECTIONS WHICH HAVE A JET
      READ(LUN, 10, END=60 ) (DCMU(K),K=1,NROWSJ)
   10 FORMAT(8F10.6)
   20 IF(DCMU(1) .LT. 800.0) GO TO 30
      IR = 2
      RETURN
C
C  REARRANGE THE DATA INTO THE PROPER SEQUENCE
   30 KP = 0
      DO 50 K = 1,NROWS
   40 CMU(K) = 0.00
      IF(NJ(K) .EQ. 0) GO TO 50
      KP = KP + 1
      CMU(K) = DCMU(KP)
   50 CONTINUE
      IR = 1
      RETURN
C
C  AN END OF FILE HAS BEEN READ.  THIS RUN IS COMPLETELY FINISHED.
   60 WRITE(6, 70 )
   70 FORMAT(1H1///41X,37HNO MORE CMU CASES HAVE BEEN REQUESTED)
      IR = 3
      RETURN
      END
      SUBROUTINE STAGE2
C
C  THIS PROGRAM CONTROLS THE FORMATION AND SOLUTION OF
C  THE SYSTEM OF LINEAR EQUATIONS
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C  FORM THE SYSTEM OF LINEAR EQUATIONS
   10 CALL STG2D
C
C  SOLVE THE SYSTEM OF LINEAR EQUATIONS
   20 CALL STG2S
      IF(IR .EQ. 2) GO TO 30
C
C  THE SOLUTION HAS NOW BEEN COMPLETED.   RETURN NORMALLY TO MAIN.
C  *** HALLELULIAH ***
      IR = 1
      GO TO 40
C
C*****************************************************************
C  A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.
   30 IR = 2
   40 RETURN
C*****************************************************************
C
      END
      SUBROUTINE STG2D
C
C  THIS PROGRAM CONTROLS THE CALCULATION OF ALL EVD DOWNWASH
C  INFLUENCE COEFFICIENTS AND THE FORMATION OF THE LEFT AND RIGHT SIDE
C  MATRICES.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      DIMENSION W(610)
C
C  IF THIS IS A NEW CMU CASE, AUGMENT THE EXISTING DOWNWASH MATRIX ROWS
C  ON THE JET
      ISIZE = NEWMAX
      IF(NEWCMU .EQ. 1) GO TO 10
      IF(NEWMAX .GT. NWT) CALL SHUFL2(W,ISIZE,NEWMAX)
      GO TO 30
C
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENTS AT ALL CONTROL POINTS
C  DUE TO ALL TRIANGULAR, LEADING EDGE AND FAR-JET EVD ELEMENTS
   10 ISIZE = NMAX
      IF((LOGIC .EQ. 2) .AND. (ISYMM .GT. 0)) GO TO 30
      CALL DWNWSH(W,ISIZE)
C
C  AUGMENT THE MATRIX ROWS FOR CONTROL POINTS ON THE JET.
C  NOTE THAT THIS MUST BE DONE EVEN THOUGH CMU MAY BE 0.0,
C  IN ORDER TO PREPARE FOR FUTURE NONZERO CMU CASES.
   20 IF(NMAX .GT. NWT) CALL SHUFL1(W,ISIZE)
C
C  DEVELOP THE RIGHT SIDE COLUMN MATRIX
   30 ISIZE = NEWMAX
      DO 80 N = 1,NCASES
      LCASE = N
C
C  DEFINE THE LCASE COLUMN, NOT INCLUDING THE INFLUENCE OF ANY HINGES
   40 CALL COLUM1(LCASE)
C
C  COMPUTE THE HINGE DOWNWASH INFLUENCE FACTORS
      IF((LCASE .EQ. 1) .OR. (IHINGE .EQ. 0)) GO TO 80
      DO 50 I = 1,NEWMAX
      W(I) = 0.00
   50 CONTINUE
   60 CALL HINGE(W,ISIZE,NEWMAX,LCASE)
C
C  MODIFY THE LCASE COLUMN TO INCLUDE THE INFLUENCE OF ALL HINGES
   70 CALL COLUM2(W,ISIZE,NEWMAX,LCASE)
```

```
   80 CONTINUE
C
C  THE MATRIX DEVELOPMENT IS NOW COMPLETE.
C  PUT THE MATRIX SYSTEM IN THE PROPER FORM FOR SOLUTION.
   90 ISIZE = NEWMAX + NCASES
  100 CALL PREP(W,ISIZE,NEWMAX)
      RETURN
      END
      SUBROUTINE DWNWSH(W,ISIZE)
C
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENT MATRIX.
C  THE MATRIX IS STORED ON THE DIRECT ACCESS SCRATCH FILE.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1              D(40),KK(600),ITYPE(600)
      DIMENSION W(ISIZE)
C
C
C  COMPUTE ALL THE DOWNWASH COEFFICIENTS
C
      IWRITE = 0
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,38X,44HWING - DUE - TO - WING - AND - JET  DOWNWASH/)
C
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET
      DO 190 I = 1,NMAX
      IWRITE = IWRITE + 1
C       FIND(1'IWRITE)                       *** COMMENTED OUT BY JAC **
      KI = KK(I)
C
C  CYCLE THE VORTEX POINTS ON THE WING AND JET
      DO 150 J = 1,NMAX
C  COMPUTE THE GENERAL G OMETRIC PARAMETERS
      KJ = KK(J)
   20 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)
      YY = Y(KI) - Y(KJ)
      IT = ITYPE(J)/10
C
C  DECIDE WHICH EVD TYPE TO USE.  ONLY THE TRIANGULAR PART OF HINGES
C  WILL BE CONSIDERED AT THIS TIME.
      GO TO ( 50 , 80 , 110 , 50 ), IT
   30 WRITE(6, 40 ) J,IT
   40 FORMAT(1H ,35X,21H** WARNING ** ELEMENT,I4,19H HAS AN ITYPE VALUE,
     1      3H OF,I3/39X,42HAN EQUIVALENT TRIANGULAR DOWNWASH WAS USED)
C
C  REGULAR TRIANGULAR EVD  (INCLUDING TRIANGULAR PART OF HINGE EVD)
   50 D1 = DEL(J-1) * CHORD(KJ)
      IN1 = IW(KJ) + NW(KJ) - 1
      IF(J .EQ. IJ(KJ))D1 = DEL(IN1) * CHORD(KJ)
      D2 = DEL(J) * CHORD(KJ)
   60 W(J) = EVD1(X,YY,D1,D2,DELTA(KJ))
C  SUPERIMPOSE THE DOWNWASH FROM THE LEFT SIDE OF THE WING IF THIS IS A
C  SYMETRIC OR ANTI-SYMETRIC CASE
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
   70 WDUMMY = EVD1(X,YY,D1,D2,DELTA(KJ))
      IF(ISYMM .LT. 0)WDUMMY = -WDUMMY
      GO TO 140
C
C  LEADING EDGE EVD
   80 D2 = DEL(J) * CHORD(KJ)
   90 W(J) = EVD2(X,YY,D2,DELTA(KJ))
C  SUPERIMPOSE DOWNWASH
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
  100 WDUMMY = EVD2(X,YY,D2,DELTA(KJ))
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY
      GO TO 140
C
C  FAR - JET EVD
  110 D1 = DEL(J-1) * CHORD(KJ)
  120 W(J) = EVD3(X,YY,D1,D(KJ),DELTA(KJ))
C  SUPERIMPOSE DOWNWASH
      IF(ISYMM .GT. 0) GO TO 150
      YY = Y(KI) + Y(KJ)
  130 WDUMMY = EVD3(X,YY,D1,D(KJ),DELTA(KJ))
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY
  140 W(J) = W(J) + WDUMMY
  150 CONTINUE
C
C  STORE THE DOWNWASH AT CONTROL POINT I ON THE DIRECT ACCESS UNIT
C
  160 WRITE(1'IWRITE) W
      IF(IPRINT .GE. 0) GO TO 190
      IF(I .EQ. NWT+1) WRITE(6, 170 )
  170 FORMAT(1H1,38X,43HJET - DUE - TO - WING - AND - JET  DOWNWASH/)
      WRITE(6, 180 ) I,W
  180 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))
  190 CONTINUE
      RETURN
C
C  THE DIRECT ACCESS UNIT NOW CONTAINS THE FOLLOWING -
C     WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NWT RECORDS)
C     JET -DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NJT RECORDS)
      END
      FUNCTION EVD1(X,Y,D1,D2,DELTA)
C
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
```

```
C   DUE TO A REGULAR TRIANGULAR EVD ELEMENT WITH UNIT PEAK VORTICITY,
C   LOCATED AT THE ORIGIN 0,0
C
      R(A,B) = SQRT(A*A + B*B)
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS
      IF(Y .LT. 0.0) Y = -Y
      YMD = Y - DELTA
      YPD = Y + DELTA
      PART1 = (D1 + D2) * (1.0/YMD - 1.0/YPD)
      IF(X/(0.50*(D1+D2)) .GT. 100.0) GO TO 90
   10 XPD = X + D1
      XMD = X - D2
   20 ROP = R(X,YMD)
      R1P = R(XPD,YMD)
      R2P = R(XMD,YMD)
   30 ROPP = R(X,YPD)
      R1PP = R(XPD,YPD)
      R2PP = R(XMD,YPD)
C
C   CALCULATE THE DOWNWASH
   40 PART2 = (XPD/D1) * ((R1P-ROP)/YMD - (R1PP-ROPP)/YPD)
      PART3 = (XMD/D2) * ((R2P-ROP)/YMD - (R2PP-ROPP)/YPD)
      YRATIO = (YPD+ROPP) / (YMD+ROP)
   50 PART4 = (XPD/D1) * ALOG(((YMD+R1P)/(YPD+R1PP)) * YRATIO)
      PART5 = (XMD/D2) * ALOG(((YMD+R2P)/(YPD+R2PP)) * YRATIO)
   60 PART6 = YMD * ALOG((XPD+R1P)/(X+ROP))
      PART7 = YPD * ALOG((XPD+R1PP)/(X+ROPP))
   70 PART8 = YMD * ALOG((XMD+R2P)/(X+ROP))
      PART9 = YPD * ALOG((XMD+R2PP)/(X+ROPP))
   80 EVD1 = -(PART1 + (PART2 + PART3) - 2.0*(PART4 + PART5)
     1     -((PART6 - PART7)/D1 + (PART8 - PART9)/D2)) / 25.13274
      RETURN
   90 EVD1 = -PART1 / 12.56673
  100 RETURN
      END
      FUNCTION EVD2(X,Y,DEL,DELTA)
C
C   THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
C   DUE TO A LEADING EDGE EVD ELEMENT WITH UNIT AVERAGE VORTICITY,
C   LOCATED AT THE ORIGIN 0,0
C
      DIMENSION SI(9),FACTOR(9)
C
      R(A,B) = SQRT(A*A + B*B)
      G(A) = 1.00/SQRT(A) - A
      S(A) = ABS(A) / A
C
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,
     1        0.3242534, 0.6133714, 0.8360311, 0.9681602/
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,
     1            0.3123471,0.2606107,0.1806482,0.0812744/
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS
   10 XB = X / DEL
      YB = Y / DEL
      DB = DELTA / DEL
      YPD = YB + DB
      YMD = YB - DB
   20 IF(XB .GT. 100.0) GO TO 280
      XM1 = XB - 1.00
   30 ROP = R(XB,YMD)
   40 ROPP = R(XB,YPD)
C
C   CALCULATE RK(XB)
      IF(ABS(XB) .LT. 1.0E-04) GO TO 100
      IF(ABS(XM1) .LT. 1.0E-06) GO TO 80
   50 PART = ALOG(ABS(XM1/XB))
      PART1 = XB * PART + 1.00
      IF(XB .GT. 0.00) GO TO 70
   60 SQX = SQRT(-XB)
      RK = -2.00 / SQX * ATAN(1.00/SQX) + PART1
      GO TO 90
   70 SQX = SQRT(XB)
      RK = -ALOG(ABS((1.00-SQX) /(1.00+SQX))) / SQX + PART1
      GO TO 90
   80 RK = 2.386294
C
C   CALCULATE P(XB)
   90 PART2 = ROP/YMD - ROPP/YPD
      P = PART2 * RK
      GO TO 110
  100 P = 0.00
C
C   CALCULATE F(XB) BY GAUSSIAN INTEGRATION.
  110 IF((XB .GT. 0.0) .AND. (XB .LT. 1.00)) GO TO 150
C   XB IS NOT WITHIN THE X DIMENSIONS OF THE ELEMENT.
      F = 0.00
      DO 140 N = 1,9
  120 SB = (SI(N)+1.00) / 2.00
      XMS = XB - SB
      GS = G(SB)
  130 PART3 = (GS*(R(XMS,YMD)-ROP))/YMD - (GS*(R(XMS,YPD)-ROPP))/YPD
      F = F + FACTOR(N) * PART3 / XMS
  140 CONTINUE
      F = 0.50 * F
      GO TO 270
C
C   XB IS WITHIN THE X DIMENSIONS OF THE ELEMENT.   CALCULATE F0.
  150 F0 = 0.00
      GPX = 0.00
```

```
      GPPX = 0.00
      IF(XB .LT. 1.0E-04) GO TO 170
      GX = G(XB)
      GPX = GX * (ABS(YMD) - ROP)
      GPPX = GX * (ABS(YPD) - ROPP)
      IF(XM1 .GT. -1.0E-06) GO TO 170
  160 F0 = -(GPX/YMD - GPPX/YPD) * PART
C  CALCULATE F1 BY GAUSSIAN INTEGRATION.
  170 F1 = 0.00
      DO 250 N = 1,9
  180 SB = (SI(N)+1.00) / 2.00
      XMS = XB - SB
  190 IF(ABS(XMS) .LT. 1.0E-04) GO TO 220
      GS = G(SB)
  200 PART4 = (GS*(R(XMS,YMD)-ROP) - GPX) / XMS
  210 PART5 = (GS*(R(XMS,YPD)-ROPP) - GPPX) / XMS
      PART6 = PART4/YMD - PART5/YPD
      GO TO 240
  220 PART4 = S(YMD) - S(YPD) - PART2
      PART5 = 1.00 + 0.50 / (SQRT(XB))**3
  230 PART6 = PART4 * PART5
  240 F1 = F1 + FACTOR(N) * PART6
  250 CONTINUE
  260 F = F0 + 0.50 * F1
C
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT.
  270 EVD2 = -(1.50 * (1.00/YMD - 1.00/YPD) + P + F) / 18.84956
      RETURN
  280 EVD2 = -(1.00/YMD - 1.00/YPD) / 6.283185
      RETURN
      END
      FUNCTION EVD3(X,Y,DEL,D,DELTA)
C
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y DUE TO A
C  FAR-JET EVD ELEMENT WITH UNIT PEAK VORTICITY, LOCATED AT THE ORIGIN
C
      R(A,B) = SQRT(A*A + B*B)
C
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS
      IF(Y .LT. 0.0) Y = -Y
      YMD = Y - DELTA
      YPD = Y + DELTA
   10 XPD = X + D
   20 PART1 = (DEL/2.00 + D) * (1.00/YMD - 1.00/YPD)
C  CHECK ON INFINITY
      IF((XPD/YMD)**2 .GT. 1.0E06) GO TO 160
      XPD1 = X + DEL
      XD = X/DEL
      ROP = R(X,YMD)
      ROPP = R(X,YPD)
   30 R1P = R(XPD1,YMD)
      R1PP = R(XPD1,YPD)
      RDP = R(XPD,YMD)
      RDPP = R(XPD,YPD)
C
C  CALCULATE F0
   40 PART2 = ROP/YMD - ROPP/YPD
      PART3 = 0.50 * (XD+1.00) * ((R1P-ROP)/YMD - (R1PP-ROPP)/YPD)
   50 PART4 = ALOG(ABS(((YMD+R1P)/(YPD+R1PP)) * ((YPD+ROPP)/(YMD+ROP))))
      PART5 = YMD/DEL * ALOG((XPD1+R1P)/(X+ROP))
   60 PART6 = YPD/DEL * ALOG((XPD1+R1PP)/(X+ROPP))
   70 F0 = PART1 - 0.50*PART2 + PART3 - (XD+1.00)*PART4
     1        - 0.50*(PART5-PART6)
C
C  CALCULATE F1
      IF(ABS(XPD/D) .LT. 1.0E-02) GO TO 130
C  X IS NOT NEAR -D
      Q = D/XPD
   80 PART1 = -D * Q * (1.00/YMD - 1.00/YPD)
      PART2 = Q * PART2
   90 PART3 = Q * Q * ALOG(ABS((YMD+ROP)/(YPD+ROPP)))
  100 PART4 = YMD/RDP*ALOG(ABS((-XPD+RDP*((ROP+RDP)/D))/(RDP-XPD)))
  110 PART5 = YPD/RDPP*ALOG(ABS((-XPD+RDPP*((ROPP+RDPP)/D))/(RDPP-XPD)))
  120 F1 = PART1 + PART2 - PART3 + Q * Q * (PART4 - PART5)
      GO TO 150
C  X IS NEAR -D
  130 X = -D
      ROP = R(X,YMD)
      ROPP = R(X,YPD)
      PART2 = ROP/YMD - ROPP/YPD
      PART3 = (X/YMD)**2 * ALOG(ABS((YMD+ROP)/X))
      PART4 = (X/YPD)**2 * ALOG(ABS((YPD+ROPP)/X))
  140 F1 = -0.50*PART2 - 0.50*(PART3 - PART4)
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT
C
  150 EVD3 = -(F0 + F1) / 12.56637
      RETURN
  160 EVD3 = -PART1 / 6.283185
      RETURN
      END
      FUNCTION EVD4(X,Y,D1,D2,DELTA)
C
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y
C  DUE TO A HINGE EVD ELEMENT WITH ONE RADIAN TURNING ANGLE
C  LOCATED AT THE ORIGIN 0,0
C
      DIMENSION SI(9),FACTOR(9)
      R(A,B) = SQRT(A*A + B*B)
      CHANGE(A,B,C) = 0.50 * (C * (B-A) + (A+B))
      S(A) = ABS(A) / A
      SO(T) = 0.50 * (-D1L/D1*(1.00-S(T)) + D2L/D2*(1.00+S(T)))
```

```
      G(A) = ALOG(ABS(A)) - SQ(A) * A
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,
     1        0.3242534, 0.6133714, 0.8360311, 0.9681602/
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,
     1            0.3123471,0.2606107,0.1806482,0.0812744/
C
C   CALCULATE THE BASIC GEOMETRICAL PARAMETERS.
      DB = 0.50 * (D1 + D2)
   10 XB = X / DB
      XMD = X - D2
      XPD = X + D1
   20 YMD = Y - DELTA
      YPD = Y + DELTA
   30 ROP = R(X,YMD)
      ROPP = R(X,YPD)
      AX = ABS(X)
      AXB = ABS(XB)
      D1L = ALOG(D1)
      D2L = ALOG(D2)
      XD1 = X/D1
      XD2 = X/D2
   40 PART5 = 1.00/YMD - 1.00/YPD
      PART6 = -(D1+D2) + 0.50 * (D1*D1L + D2*D2L)
C
C   CALCULATE RK(X).
      IF(AXB .LT. 7.5) GO TO 70
      N = 0
      RK = 0.00
      D1X = D1/X
      D2X = D2/X
      DO 60 N1 = 1,100,2
      RKN = 0.000
      DO 50 N2 = 1,2
      N = N + 1
      RN = N
      RNN = N * (N+1)
      RKN = RKN + (-1.0)**(N+1) * (D1L/RNN - 1.0/(RN*RN)) * D1X**N
     1      +(D2L/RNN - 1.0/(RN*RN)) * D2X**N
   50 CONTINUE
      RK = RK + RKN
      IF(ABS(RKN/RK) .LT. 1.0E-07) GO TO 190
   60 CONTINUE
   70 IF(AXB .LT. 1.0E-04) GO TO 200
      SX = S(X)
      RK1 = 0.00
      RK2 = 0.00
      RK3 = 0.00
      RK4 = 0.00
      RK1P = ABS(XD1+1.00)
   80 IF(RK1P .LT. 1.0E-06) GO TO 90
      RK1 = (ALOG(ABS(XD1)) + (XD1+1.00) * D1L) * ALOG(ABS(XPD/X))
   90 RK2P = ABS(XD2-1.00)
      IF(RK2P .LT. 1.0E-06) GO TO 100
      RK2 = -(ALOG(ABS(XD2)) - (XD2-1.00) * D2L) * ALOG(ABS(XMD/X))
  100 IF(ABS(D1/AX-1.00) .LT. 1.0E-04) GO TO 140
C   CALCULATE RK3 BY GAUSSIAN INTEGRATION.
      AL = 1.00
      BL = D1/AX
      DO 130 N = 1,9
  110 T = CHANGE(AL,BL,SI(N))
  120 RK3 = RK3 + FACTOR(N) * (ALOG(T) / (SX+T))
  130 CONTINUE
      RK3 = 0.50 * (BL-AL) * RK3
C   CALCULATE RK4 BY GAUSSIAN INTEGRATION.
  140 IF(ABS(D2/AX-1.00) .LT. 1.0E-04) GO TO 180
      AL = 1.00
      BL = D2/AX
      DO 170 N = 1,9
  150 T = CHANGE(AL,BL,SI(N))
  160 RK4 = RK4 + FACTOR(N) * (ALOG(T) / (SX-T))
  170 CONTINUE
      RK4 = 0.50 * (BL-AL) * RK4
  180 RK = -2.467401 * SX - (D1L-D2L) + (RK1 + RK2) + (RK3 + RK4)
  190 P = (ROP/YMD - ROPP/YPD) * RK
      GO TO 210
  200 P = 0.00
C
C   CALCULATE F(X) BY GAUSSIAN INTEGRATION.
  210 IF((X .GT. -D1) .AND. (X .LT. D2)) GO TO 290
C   X IS NOT WITHIN THE DIMENSIONS OF THE ELEMENT.
C   LEFT SIDE INTEGRAL.
      FL = 0.00
      AL = -D1
      BL = 0.00
      DO 240 N = 1,9
  220 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      GS = G(SY)
  230 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) / YPD
      FL = FL + FACTOR(N) * PART1 / XMS
  240 CONTINUE
      FL = 0.50 * (BL-AL) * FL
C   RIGHT SIDE INTEGRAL
      FR = 0.00
      AL = 0.00
      BL = D2
      DO 270 N = 1,9
  250 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      GS = G(SY)
  260 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) / YPD
```

```
      FR = FR + FACTOR(N) * PART1 / XMS
  270 CONTINUE
      FR = 0.50 * (BL-AL) * FR
  280 F = FL + FR
      GO TO 460
C
C  X IS WITHIN THE DIMENSIONS OF THE ELEMENT
C  CALCULATE F0
  290 F0 = 0.00
      GPX = 0.00
      GPPX = 0.00
      IF(AXB .LT. 1.0E-04) GO TO 310
      SOX = SO(X) - 1.00/X
      GX = G(X)
      GPX = GX * (ABS(YMD)-ROP)
      GPPX = GX * (ABS(YPD)-ROPP)
      IF(((1.00-XD2) .LT. 1.0E-06) .OR. ((XD1+1.00) .LT. 1.0E-06))
     1     GO TO 310
  300 F0 = -(GPX/YMD - GPPX/YPD) * ALOG(ABS(XMD/XPD))
C
C  CALCULATE F1
C  LEFT SIDE INTEGRAL
  310 FL = 0.00
      AL = -D1
      BL = 0.00
      DO 370 N = 1,9
  320 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 350
      GS = G(SY)
  330 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS
  340 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS
      PART4 = PART2/YMD - PART3/YPD
      GO TO 360
  350 PART2 = ROP/YMD - ROPP/YPD
      PART4 = (S(YMD) - S(YPD) - PART2) * SOX
  360 FL = FL + FACTOR(N) * PART4
  370 CONTINUE
      FL = 0.50 * (BL-AL) * FL
C
C  RIGHT SIDE INTEGRAL
  380 FR = 0.00
      AL = 0.00
      BL = D2
      DO 440 N = 1,9
  390 SY = CHANGE(AL,BL,SI(N))
      XMS = X - SY
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 420
      GS = G(SY)
  400 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS
  410 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS
      PART4 = PART2/YMD - PART3/YPD
      GO TO 430
  420 PART2 = ROP/YMD - ROPP/YPD
      PART4 = (S(YMD) - S(YPD) - PART2) * SOX
  430 FR = FR + FACTOR(N) * PART4
  440 CONTINUE
      FR = 0.50 * (BL-AL) * FR
  450 F = FL + FR + F0
C
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT
  460 EVD4 = (PART5 * PART6 + P + F) / 19.739202
      RETURN
      END
      SUBROUTINE SHUFL1(W,ISIZE)
C
C  THIS SUBROUTINE READS THE PORTION OF THE DOWNWASH MATRIX WHICH
C  CONTAINS THE DOWNWASH DUE TO THE JET, AUGMENTS IT ACCORDING TO
C  THE CURRENT CMU VALUES, AND WRITES IT BACK ON UNIT 1
C  BEHIND THE DOWNWASH MATRIX
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION A(600),AIM1(600),W(ISIZE)
C
      NWT1 = NWT + 1
      FIND(1'NWT1)
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX/)
C
C  PREPARE THE SOLUTION MATRIX FOR ROWS ON THE JET
C
C       I IS THE COUNTER FOR IDENTIFYING ELEMENTS ON THE JET
C   IREAD IS THE COUNTER FOR DOWNWASH ROWS ON THE JET STORED ON UNIT 1
C  IWRITE IS THE COUNTER FOR AUGMENTED ROWS TO BE WRITTEN ON UNIT 1
   20 IREAD = NWT
      IWRITE = NMAX
      C1 = 0.125000
      C3 = 0.375000
      DO 150 I = NWT1,NMAX
   30 IM1 = I - 1
      IP1 = I + 1
      IREAD = IREAD + 1
      IWRITE = IWRITE + 1
      K = KK(I)
C
C  READ THE ITH ROW OF THE DOWNWASH MATRIX  (IREADTH RECORD)
```

```
   40 READ(1'IREAD) W
C  FIND THE PLACE TO WRITE THE ITH AUGMENTED ROW (IWRITETH RECORD)
C       FIND(1'IWRITE)                        ***COMMENTED OUT BY JAC ***
C
C  SAVE THE EXISTING ROW OF SIMPLE DOWNWASH COEFFICIENTS
      DO 50 J = 1,NMAX
      A(J) = W(J)
   50 CONTINUE
C
C  SUBTRACT THE PREVIOUS ROW FROM THE PRESENT ROW IF THE DOWNWASH POINT
C  IS NOT ON A LEADING JET ELEMENT
      PROD1 = CMUP(K) * DEL(I) * CHORD(K)
      PROD2 = CMUP(K) * DEL(IM1) * CHORD(K)
   60 IF(I .EQ. IJ(K)) GO TO 90
      DO 70 J = 1,NMAX
      W(J) = W(J) - AIM1(J)
   70 CONTINUE
C
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS FURTHER
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 100
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT
   80 W(IM1) = W(IM1) + C1 * PROD2
      W(I) = W(I) + C3 * (PROD1 + PROD2)
      W(IP1) = W(IP1) + C1 * PROD1
      GO TO 110
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT
   90 W(I) = W(I) + C3 * PROD1
      W(IP1) = W(IP1) + C1 * PROD1
      GO TO 110
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT
  100 W(IM1) = W(IM1) + C1 * PROD2
      W(I) = W(I) + C3 * PROD2 + CMUP(K) * D(K)
C
C  STORE THE AUGMENTED ITH ROW ON UNIT 1  (IWRITETH RECORD)
  110 WRITE(1'IWRITE) W
C  FIND THE PLACE TO READ THE NEXT DOWNWASH ROW (IREAD+1ST RECORD)
      FIND(1'IREAD+1)
C  PRINT THE AUGMENTED PORTION OF THE MATRIX
      IF(IPRINT .LT. 0) WRITE(6, 120 ) I,W
  120 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))
C
C  SAVE THE ITH ROW FOR USE AS THE I-1 ROW ON THE NEXT PASS
  130 DO 140 J = 1,NMAX
      AIM1(J) = A(J)
  140 CONTINUE
  150 CONTINUE
      RETURN
C
C  DIRECT ACCESS UNIT 1 NOW CONTAINS THE FOLLOWING -
C    WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NWT RECORDS)
C     JET-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NJT RECORDS)
C     JET-DUE-TO-WING-AND-JET AUGMENTED DOWNWASH COEFFICIENTS (NJT RECS)
      END
      SUBROUTINE SHUFL2(W,ISIZE,NEWMAX)
C
C  THIS SUBROUTINE READS EACH MATRIX ROW CORRESPONDING TO A DOWNWASH
C  CONTROL POINT ON THE JET, MODIFIES IT ACCORDING TO THE NEW VALUES
C  OF CMU, AND RESTORES IT IN ITS ORIGINAL PLACE
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION W(ISIZE)
C
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX)
C
C  CYCLE THE AUGMENTED MATRIX ROWS
      IREAD = NMAX
      NWT1 = NWT + 1
      DO 100 I = NWT1,NEWMAX
      IREAD = IREAD + 1
      FIND(1'IREAD)
   20 IM1 = I - 1
      IP1 = I + 1
      K = KK(I)
   30 CMUDIF = (CMUP(K) - CMUPP(K)) * CHORD(K)
C
C  READ THE ITH AUGMENTED MATRIX ROW
   40 READ(1'IREAD) W
      FIND(1'IREAD)
C
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS ACCORDING TO THE NEW CMU
      IF(I .EQ. IJ(K)) GO TO 60
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 70
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT
   50 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF
      W(I) = W(I) + 0.3750 * (DEL(IM1)+DEL(I)) * CMUDIF
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF
      GO TO 80
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT
   60 W(I) = W(I) + 0.3750 * DEL(I) * CMUDIF
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF
      GO TO 80
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT
   70 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF
```

```
      W(I) = W(I) + (0.3750*DEL(IM1) + D(K)/CHORD(K)) * CMUDIF
C
C  WRITE THE REVISED ITH ROW ON UNIT 1
   80 WRITE(1'IREAD) W
      IF(IPRINT .LT. 0) WRITE(6, 90 ) I,W
   90 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))
  100 CONTINUE
      RETURN
      END
      SUBROUTINE COLUM1(LCASE)
C
C  THIS SUBROUTINE SETS UP THE RIGHT SIDE COLUMN MATRIX WITHOUT
C  CONSIDERATION OF ANY HINGE DOWNWASH INFLUENCE
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/SOLV1/B(600,10)
C
C  DEFINE THE ELEMENTS ON THE WING
      DO 10 I = 1,NWT
      B(I,LCASE) = EPS(I,LCASE) / 57.295779
   10 CONTINUE
C
C  DEFINE THE ELEMENTS ON THE JET
      I = NWT
      DO 40 K = 1,NROWS
      NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 40
C  FIRST JET ELEMENT
      I = I + 1
      KKI = KK(I)
   20 B(I,LCASE) = THETA(KKI,LCASE) / 57.295779
C  REMAINING JET ELEMENTS
      DO 30 L = 2,NJK
      I = I + 1
      B(I,LCASE) = 0.00
   30 CONTINUE
   40 CONTINUE
      RETURN
      END
      SUBROUTINE COLUM2(H,ISIZE,NEWMAX,LCASE)
C
C  THIS SUBROUTINE ADDS THE APPROPRIATE HINGE DOWNWASH INFLUENCE
C  TO THE RIGHT SIDE COLUMN MATRIX
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/B(600,10)
      DIMENSION H(ISIZE)
C
C  DEFINE THE ELEMENTS ON THE WING
      DO 10 I = 1,NEWMAX
      B(I,LCASE) = B(I,LCASE) - H(I)
   10 CONTINUE
C
C  DEFINE THE ELEMENTS ON THE JET
      IF(NEWMAX .EQ. NWT) RETURN
      I = NWT
      DO 90  K = 1,NROWS
   20 NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 90
C  COMPUTE THE CMU INFLUENCE FACTORS H1 AND H2
      I = I + 1
      H1 = 0.00
      H2 = 0.00
      BTA = BETA(I,LCASE)
      IF(ABS(BTA) .LT. 0.0001) GO TO 50
      BTA = BTA / 57.295779
   30 D2 = DEL(I) * CHORD(K)
      DL = ALOG(D2)
      PROD = -CMUP(K) * D2 * BTA / 3.1415927
   40 H1 = PROD * (1.6931472 - 0.750 * DL)
      H2 = PROD * (0.3068528 - 0.250 * DL)
C  FIRST POINT ON THE JET
   50 B(I,LCASE) = B(I,LCASE) + H1
C  SECOND POINT ON THE JET
      I = I + 1
   60 B(I,LCASE) = B(I,LCASE) + H(I-1) + H2
C  REMAINING POINTS ON THE JET
      IF(NJK .LT. 3) GO TO 90
      DO 80 L = 3,NJK
      I = I + 1
   70 B(I,LCASE) = B(I,LCASE) + H(I-1)
   80 CONTINUE
   90 CONTINUE
      RETURN
      END
      SUBROUTINE HINGE(H,ISIZE,NEWMAX,LCASE)
C
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENTS
C  AT EACH DOWNWASH CONTROL POINT DUE TO ALL DEFLECTED HINGE ELEMENTS.
C  FOR EACH CONTROL POINT THE INFLUENCE COEFFICIENTS ARE MULTIPLIED BY
C  THEIR RESPECTIVE DEFLECTION ANGLE AND SUMMED UP TO OBTAIN THE
C  COMPLETE HINGE-INDUCED DOWNWASH.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
```

```
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1            D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION H(ISIZE)
C
      IF(LCASE .GT. 2) GO TO 20
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1)
      ILINES = 1
C
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET
   20 DO 110 I = 1,NEWMAX
      KI = KK(I)
C
C  CYCLE THE VORTEX POINTS ON THE WING AND JET
      DO 100 J = 1,NEWMAX
C  CHECK WHETHER THERE IS A DEFLECTED HINGE AT ELEMENT J
   30 B = BETA(J,LCASE)
      IF(ABS(B) .LT. 0.0001) GO TO 100
      B = B / 57.295779
C  COMPUTE THE GEOMETRIC PARAMETERS
      KJ = KK(J)
   40 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)
      YY = Y(KI) - Y(KJ)
   50 D2 = DEL(J) * CHORD(KJ)
      D1 = DEL(J-1) * CHORD(KJ)
      IN1 = IW(KJ) + NW(KJ) - 1
   60 IF(ITYPE(J) .EQ. 43) D1 = DEL(IN1) * CHORD(KJ)
C  COMPUTE AND SUM UP THE INFLUENCE OF ELEMENT J
   70 H(I) = H(I) + EVD4(X,YY,D1,D2,DELTA(KJ)) * B
C  SUPERIMPOSE DOWNWASH FOR SYMETRIC OR ANTI-SYMETRIC GEOMETRY
      IF(ISYMM .GT. 0) GO TO 100
      YY = Y(KI) + Y(KJ)
   80 HDUMMY = EVD4(X,YY,D1,D2,DELTA(KJ))
      IF(ISYMM .LT. 0) HDUMMY = - HDUMMY
   90 H(I) = H(I) + HDUMMY * B
  100 CONTINUE
  110 CONTINUE
C
C  PRINT OUT THE DOWNWASH IF REQUIRED
      IF(IPRINT .GE. 0) RETURN
      NEXT = NEWMAX/10 + 3
      IF(((ILINES+NEXT) .LT. 56) .OR. (ILINES .EQ. 1)) GO TO 120
      WRITE(6, 10 )
      ILINES = 1
  120 IF(IPRINT .LT. 0) WRITE(6, 130 ) LCASE,H
  130 FORMAT(1H0,35X,44HHINGE INFLUENCE COEFFICIENTS FOR FUNDAMENTAL,
     1          5H CASE, I3,60(/1X,10E13.5))
      ILINES = ILINES + NEXT
      RETURN
      END
      SUBROUTINE STG2S
C
C  THIS PROGRAM CONTROLS SOLUTION OF THE MATRIX SYSTEM
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/SOLV1/GAMMA(600,10)
      COMMON/SOLV2/WKAREA(10000)
      DIMENSION W(600),SUMMER(600)
C
      IF(IPRINT .LT. 0) WRITE(6, 10 )
   10 FORMAT(1H1,53X,14HGAMMA SOLUTION)
C
C  SOLVE THE MATRIX FOR GAMMA USING MATRIX
      NSIZE = 10000
      NIN = 2
      NSCR1 = 3
      NSCR2 = 4
   20 CALL MATRIX(NEWMAX,NCASES,NSIZE,NIN,NSCR1,NSCR2,IR)
      IF(IR .EQ. 2) GO TO 90
C  MATRIX HAS STORED THE SOLUTION IN THE FIRST STORAGE LOCATIONS OF THE
C  WKAREA ARRAY.  TRANSFER THIS DATA INTO THE GAMMA ARRAY.
      ISUM = 0
   30 DO 70 N = 1,NCASES
      DO 40 J = 1,NEWMAX
      GAMMA(J,N) = WKAREA(ISUM+J)
   40 CONTINUE
      IF(IPRINT .LT. 0) WRITE(6, 50 ) N,(GAMMA(J,N),J=1,NEWMAX)
   50 FORMAT(1H0,50X,16HFUNDAMENTAL CASE,I4,60(/1X,10E13.5))
   60 ISUM = ISUM + NEWMAX
   70 CONTINUE
C
C  IF REQUIRED, CHECK THE SOLUTION BY BACK SUBSTITUTION
   80 IF(IPRINT .LT. 0) CALL BAKSUB(W,SUMMER,NEWMAX)
      IR = 1
      GO TO 110
C
C  PRINT THE MATRIX ERROR MESSAGE
   90 WRITE(6, 100 )
  100 FORMAT(1H1,40X,40HMATRIX DOES NOT HAVE ENOUGH CORE TO WORK/
     1          45X,29HTHIS CASE HAS BEEN TERMINATED)
      IR = 2
  110 RETURN
      END
      SUBROUTINE PREP(TRANS,ISIZE,NEWMAX)
C
C  THIS SUBROUTINE PREPARES THE FINAL MATRIX FOR SOLUTION BY
C  CONCATINATING IT WITH THE RIGHT SIDE MATRIX AND STORING IT
```

```
C   ON SCRATCH UNIT 2 FOR INPUT TO MATRIX.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SOLV1/B(600,10)
      DIMENSION TRANS(ISIZE)
C
      IREAD = 0
      FIND(1'IREAD+1)
      REWIND 2
C
C   READ THE MATRIX COEFFICIENTS
      DO 40  I = 1,NEWMAX
      IF(I .EQ. NWT+1) IREAD = NEWMAX
      IREAD = IREAD + 1
      IF(ISIZE .GT. 600) GO TO 5
      READ(1'IREAD) TRANS
      GO TO 10
    5 READ(1'IREAD) (TRANS(J),J=1,NEWMAX)
C
C   PULL OUT THE RIGHT SIDE MATRIX COEFFICIENTS AND CONCATINATE THEM
   10 DO 20 N = 1,NCASES
      TRANS(NEWMAX+N) = B(I,N)
   20 CONTINUE
C   THE MATRIX ROW HAS NOW BEEN ASSEMBLED AND FILLS THE TRANS ARRAY.
C   WRITE THE CONCATINATED ROW ON ON UNIT 2 FOR INPUT TO MATRIX.
   30 WRITE(2) TRANS
   40 CONTINUE
C   THE SYSTEM OF LINEAR EQUATIONS IS NOW READY FOR SOLUTION
      RETURN
      END
      SUBROUTINE MATRIX(ND, MD, KD, NI, MM, NO, IR)
C
C              D I R E C T   M A T R I X   S O L U T I O N
C
      COMMON/SOLV2/A(10000)
      LOGICAL  LAST
C
      N = ND
      M = MD
      KORE = KD
      NPM = N + M
      IF (MAX0(3 * NPM, M * N) .LT. KORE) GO TO 20
      IR = 2
      RETURN
   20 MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      MP1 = M + 1
      NN = N
      NEL = NPM
C
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'
C
   30 K = (KORE - NEL) / NEL
C
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE
C
      LAST = K .GE. NN
      IF (LAST) K = NN
C
C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX
C
   40 NT = 0
      DO 50 IB = 1, K
      NS = NT + 1
      NT = NT + NEL
   50 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROW
C
      IF (K .EQ. 1) GO TO 110
C
C - - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION
C
      NELP1 = NEL + 1
      NS = - NEL
      NELP2 = NELP1 + 1
C
C - - FORM THE 'TRAPEZOIDAL' ARRAY  (8)
C
      DO 60 IB = 2, K
      NP = NELP2 - IB
      NS = NS + NELP1
      NT = NS
      DO 60 IO = IB, K
      NT = NT + NEL
      MN = NT
      NB = NS
      A(NT) = (-A(NT)) / A(NS)
      DO 60 NF = 2, NP
      MN = MN + 1
      NB = NB + 1
   60 A(MN) = A(MN) + A(NT) * A(NB)
      IF (LAST) GO TO 110
C
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE
C
```

```
      NT = 0
      NP = NEL
      NS = - NEL
      DO 70 IO = 1, K
      NS = NS + NELP1
      NT = NT + NEL
      CALL  SAVE(MT, 2, NP, NP, A(NS), 1, AA2)
   70 NP = NP - 1
      NP = NP - M
      NS = KORE - NEL + 1
C
C - - READ ANOTHER ROW
C
      DO 100 IO = 1, NP
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C
C - - MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY
C
      NT = 1
      MN = NS
      DO 90 IB = 1, K
      NB = NT
      NF = MN + 1
      A(MN) = (-A(MN)) / A(NT)
      DO 80 NN = NF, KORE
      NB = NB + 1
   80 A(NN) = A(NN) + A(MN) * A(NB)
      MN = NF
   90 NT = NT + NELP1
C
C - - WRITE THE MODIFIED ROW ON TAPE
C
      NN1 = KORE - MN + 1
  100 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)
      REWIND NOUT
      REWIND NIN
C
C - - SWITCH THE TAPES
C
      NT = NIN
      NIN = NOUT
      NOUT = NT
C
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK
C
      NEL = NEL - K
      NN = NEL - M
      GO TO 30
C
C - - REWIND ALL TAPES
C
  110 REWIND MT
      REWIND NIN
      REWIND NOUT
C
C - - CONDENSE THE MATRIX
C
      NN = NEL
      NL = NEL + 1
      IF (K .EQ. 1) GO TO 130
      NS = 1
      NT = NEL
      DO 120 IB = 2, K
      NS = NS + NELP1
      NT = NT + NEL
      DO 120 IO = NS, NT
      A(NL) = A(IO)
  120 NL = NL + 1
  130 N1 = KORE - K * M + 1
C
C - - THERE, NOW WE CAN START THE BACK-SOLUTION
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)
C
      NREM = N
      NEL = NPM
      LAST = K .EQ. N
      NPASS = 0
C
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS
C
  140 KM1 = K - 1
      KP1 = K + 1
      NS = NL - MP1
      NPASS = NPASS + 1
      DO 170 MN = 1, M
      NF = NS + MN
      A(NF) = A(NF) / A(NS)
      NT = NS
      IF (KM1 .EQ. 0) GO  TO 170
      DO 160 IB = 1, KM1
      NF = NF - IB - M
      NT = NT - MP1 - IB
      SUM = 0.0
      NP = NF
      N2 = MP1 + IB
      DO 150 IO = 1, IB
      NN = NT + IO
      NP = NP + N2 - IO
  150 SUM = SUM + A(NN) * A(NP)
  160 A(NF) = (A(NF) - SUM) / A(NT)
```

```
  170 CONTINUE
C
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)
C
      N1 = KORE + 1
      DO 190 NN = 1, K
      DO 180 MN = 1, M
      NL = NL - 1
      N1 = N1 - 1
  180 A(N1) = A(NL)
  190 NL = NL - NN
C
C - - WRITE THE SOLUTIONS ON TAPE
C
      WRITE (NIN) K
      NS = N1 - 1
      DO 200 MN = 1, M
      NT = NS + MN
  200 WRITE ( NIN ) (A(IO), IO = NT, KORE, M)
C
C - - TEST IF THIS IS THE LAST PASS
C
      IF (LAST) GO TO 280
C
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE
C
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'
C
      NELOLD = NEL
      KOLD = K
      NEL = NEL - K
      NREM = NREM - K
C
C     CALCULATE NEW K.  B AND C (REAL) WILL ALWAYS BE INTEGERS.
C     K WILL BE CALCULATED REAL AND TRUNCATED - - GOOD.
C
      B = 1 + 2*M
      C = 2*(KOLD*(M+1) - KORE)
      K = (-B + SQRT(B**2 - 4*C))/2.0
      NROW = NREM - K + 1
      IF (K .LT. NREM) GO TO 210
      LAST = .TRUE.
      NROW = 1
      K = NREM
  210 NS = 1
      NT = NELOLD + 1
C
C - - READ IN THE ROWS TO BE MODIFIED
C
      DO 270 IB = 1, NREM
      NT = NT - 1
      IF (IB .LE. NROW) GO TO 220
      NS = NS + NN
      NT = NT + NN
C
C******ADDED NEXT LINE AND MODIFIED CALL, A.P. SODERMAN, 8/10/76******
C
  220 NN=NT-NS+1
      CALL  GETT(MT, 2, NN, A(NS), 1, AA2)
      NP = N1 - 1
      NF = NT - M - KM1
      NM = NN - KOLD
      DO 240 MN = 1, M
      N2 = NF
      NA = NP + MN
      NB = NA
      SUM = 0.0
      DO 230 IO = 1, KOLD
      SUM = SUM + A(N2) * A(NA)
      N2 = N2 + 1
  230 NA = NA + M
      N2 = N2 + MN - 1
  240 A(N2) = A(N2) - SUM
C
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW
C
      NL = NT - M + 1
      IF (IB .GE. NROW) GO TO 250
      NF = NL - KP1
      NN1 = NF - NS + 1
      NN2 = NT - NL + 1
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))
      GO TO 270
  250 NF = NL - KOLD
      DO 260 MN = NL, NT
      A(NF) = A(MN)
  260 NF = NF + 1
  270 CONTINUE
      REWIND MT
      REWIND NOUT
C
C - - SWITCH THE TAPES
C
      NT = MT
      MT = NOUT
      NOUT = NT
C
C - - LOOP BACK THRU THE SOLUTION
C
```

```
      NL = NF
      GO TO 140
C
C - - START TO WRAP IT UP
C
  280 REWIND NIN
      N2 = N
C
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE
C
      DO 300 IB = 1, NPASS
      READ (NIN) K
      N1 = N2 - K + 1
      NS = N1
      NT = N2
C
C - - READ IN THE SOLUTIONS
C
      NM = NT - NS + 1
      DO 290 IO = 1, M
      CALL  GETT(NIN, 1,NM, A(NS), 1, AA2)
  290 NS = NS + N
  300 N2 = N1 - 1
      IR = 1
      RETURN
      END
      SUBROUTINE  SAVE(IU, IT, N, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO ( 10 , 20 , 30 , 40 ), IT
C
C  WRITE A1
   10 WRITE(IU) A1
      RETURN
C
C  WRITE N AND A1
   20 WRITE(IU)  N, A1
      RETURN
C
C  WRITE A1 AND A2
   30 WRITE(IU) A1, A2
      RETURN
C
C  WRITE N, A1, AND A2
   40 WRITE(IU) N, A1, A2
      RETURN
      END
      SUBROUTINE GETT(IU, IT, N1, A1, N2, A2)
C
      DIMENSION  A1(N1), A2(N2)
C
      GO TO ( 10 , 20 , 30 , 40 ), IT
C
C  READ A1
   10 READ(IU) A1
      RETURN
C
C  READ N1 AND A1
   20 READ(IU) N1, A1
      RETURN
C
C  READ A1 AND A2
   30 READ(IU) A1, A2
      RETURN
C
C  READ IDUM AND A1
   40 READ(IU) IDUM, A1
      RETURN
      END
      SUBROUTINE BAKSUB(TRANS,SUMMER,NEWMAX)
C
C  THIS SUBROUTINE BACK SUBSTITUTES THE COEFFICIENT MATRIX AND THE
C  GAMMA SOLUTION TO OBTAIN THE RIGHT SIDE MATRIX FOR THE PURPOSE OF
C  CHECKING THE MATRIX SOLUTION.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SOLV1/GAMMA(600,10)
      DIMENSION TRANS(NEWMAX),SUMMER(NEWMAX)
C
      WRITE(6, 10 )
   10 FORMAT(1H1,47X,26HBACK SUBSTITUTION SOLUTION)
C
C  CYCLE THE RIGHT HAND SIDES
      DO 60  N = 1,NCASES
      IREAD = 0
      FIND(1'IREAD+1)
C
C  CYCLE THE MATRIX ROWS CORRESPONDING TO ELEMENTS ON THE WING
      DO 40 I = 1,NEWMAX
C  READ THE COEFFICIENT MATRIX ROW
      IF(I .EQ. NWT+1) IREAD = NMAX
      IREAD = IREAD + 1
   20 READ(1'IREAD) TRANS
C
C  SUM UP THE TERMS FOR THIS ROW AND RIGHT SIDE
      SUMMER(I) = 0.00
      DO 30 J = 1,NEWMAX
      SUMMER(I) = SUMMER(I) + TRANS(J) * GAMMA(J,N)
   30 CONTINUE
```

```
   40 CONTINUE
C
C  PRINT THE NTH RIGHT SIDE COLUMN
      WRITE(6, 50 ) N,SUMMER
   50 FORMAT(1H0,50X,17HRIGHT SIDE COLUMN,I4,60(/1X,10E13.5))
   60 CONTINUE
      RETURN
      END
      SUBROUTINE STAGE3
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL LOADINGS FOR THE
C  FUNDAMENTAL AND COMPOSITE CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON /DERIV/ U0(40),CLQ,CMQ,CMQMC
      DIMENSION CPREAD(610),CP0(600),CPA(600),CPR0(600),CPRA(600),
     1          CPP(600)
      EQUIVALENCE (BETA(1,1),CP0(1)),(BETA(1,2),CPA(1)),
     1     (BETA(1,3),CPR0(1)),(BETA(1,4),CPRA(1)),(BETA(1,5),CPP(1))
C
      IF(LOGIC .EQ. 3) GO TO 80
C  CALCULATE AND PRINT THE LOADING FOR ALL FUNDAMENTAL CASES
   10 CALL STG3FC(NEWMAX)
      DO 30  N = 1,NCASES
      LCASE = N
   20 CALL STG3FS(CLQ,CMQ,CMQMC,DUM4,NEWMAX,NOALFA,LCASE)
   30 CONTINUE
      IF(LOGIC .EQ. 2) WRITE(6, 40  ) CLQ,CMQ,CMQMC
   40 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,
     1           17H ABOUT XCG, CLQ =,F10.6 /
     2           14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,
     3           33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /
     4           16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,
     5           35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)
      CALL STG3FT
C
C  CALCULATE AND PRINT THE LOADING FOR ALL COMPOSITE CASES
      IF(NCC .LT. 1) GO TO 100
      DO 60  M = 1,NCC
      MCASE = M
   50 CALL STG3C(NEWMAX,MCASE,NOALFA)
   60 CONTINUE
   70 GO TO 100
C
C  CALCULATE AND PRINT THE COEFFICIENTS AND DERIVATIVES FOR ALL
C  FUNDAMENTAL CASES
   80 CALL FUNDER(EPS,CP0,CPA,CPR0,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,AREA,
     1       CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,
     2       NROWS,ISYMM,XLEAD,TANLE,XMC)
C
C  CALCULATE AND PRINT THE STABILITY DERIVATIVES FOR ALL COMPOSITE CASES
   90 CALL COMDER(EPS,CP0,CPA,CPR0,CPRA,CPP,CPREAD,DEL,CHORD,Y,CMU,
     1       DELTA,AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,
     2       NCASES,NROWS,ISYMM,XLEAD,TANLE,XMC)
  100 RETURN
      END
      SUBROUTINE STG3FC(NEWMAX)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE LOADING
C  FOR FUNDAMENTAL CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      DIMENSION XBB(5),CPEXP(5,10)
C
C  CALCULATE AND PRINT THE CHORDWISE LOADING OF THE FUNDAMENTAL CASES
C
C  INITIALIZE THE UNUSED VALUES OF CP FOR PRINTING
      NC1 = NCASES + 1
      IF(NC1 .GT. 10) GO TO 30
      DO 20 N = NC1,10
      DO 10 I = 1,NEWMAX
      CP(I,N) = 0.00
   10 CONTINUE
   20 CONTINUE
   30 I = 0
      II = NWT
      DO 320 K = 1,NROWS
      IF(IPRINT .GT. 0) GO TO 70
      WRITE(6, 40 )
   40 FORMAT(1H1,35X,12(4H****),1H*/
     1           36X,49H*  CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  */
     2           36X,12(4H****),1H*)
      WRITE(6, 50 ) K,Y(K),CHORD(K),(N,N=1,10)
   50 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/
```

```
     1        8X,1HI,5X,2HXB,3X,9(5X,4HCASE,I2),5X,4HCASE,I3)
C
C  ON THE WING
      WRITE(6, 60 )
   60 FORMAT(1H+,4HWING)
   70 NWK = NW(K)
      DO 100 L = 1,NWK
      I = I + 1
      DO 80 N = 1,NCASES
      CP(I,N) = 2.00 * CP(I,N)
   80 CONTINUE
      IF(IPRINT .LT. 1) WRITE(6, 90 ) I,XB(I),(CP(I,N),N=1,10)
   90 FORMAT(1H ,I8,11F11.6)
  100 CONTINUE
      ILINES = NWK + 4
C
C  ON THE JET
      NJK = NJ(K)
      IF(CMU(K) .LT. 0.0001) GO TO 140
      IF(IPRINT .LT. 1) WRITE(6, 110 )
  110 FORMAT(1H ,1X,3HJET)
      DO 130 L = 1,NJK
      II = II + 1
      DO 120 N = 1,NCASES
      CP(II,N) = 2.00 * CP(II,N)
  120 CONTINUE
      IF(IPRINT .LT. 1) WRITE(6, 90 ) II,XB(II),(CP(II,N),N=1,10)
  130 CONTINUE
      ILINES = ILINES + NJK + 1
C
C  PRINT THE DETAILED LOADING ON THE SINGULAR ELEMENTS
C
C  LEADING EDGE
  140 IF(IPRINT .GT. 0) GO TO 320
      WRITE(6, 150 )
  150 FORMAT(1H0,45X,29HDETAILED LEADING EDGE LOADING)
      IP = IW(K)
      DO 160 N = 1,NCASES
      LCASE = N
      CALL EXPLE(LCASE,CP(IP,LCASE),CP(IP+1,LCASE),DEL(IP),XBB,CPEXP)
  160 CONTINUE
      DO 170 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  170 CONTINUE
      ILINES = ILINES + 7
C
C  HINGES
      IF(IHINGE .EQ. 0) GO TO 320
      J = IW(K) - 1
      DO 250 L = 1,NWK
      J = J + 1
      IF(ITYPE(J) .LT. 40) GO TO 250
      DO 180 N = 1,NCASES
      LCASE = N
      CALL EXPH1(LCASE,CP(J,N),CP(J-1,N),DEL(J-1),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  180 CONTINUE
      IF(ILINES .LT. 46) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H1)
      ILINES = 1
  200 WRITE(6, 210 ) J
  210 FORMAT(1H0,42X,33HDETAILED HINGE LOADING ON ELEMENT,I4)
      DO 220 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  220 CONTINUE
      DO 230 N = 1,NCASES
      LCASE = N
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  230 CONTINUE
      DO 240 M = 6,10
      WRITE(6, 90 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)
  240 CONTINUE
      ILINES = ILINES + 12
  250 CONTINUE
      IF((NJ(K) .EQ. 0) .OR.(CMU(K) .LT. 0.0001)) GO TO 320
      J = IJ(K)
      IF(ITYPE(J) .NE. 43) GO TO 320
      DO 260 N = 1,NCASES
      LCASE = N
      I1 = IW(K) + NW(K) - 1
      CALL EXPH1(LCASE,CP(J,N),CP(I1,N),DEL(I1),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  260 CONTINUE
      IF(ILINES .LT. 46) GO TO 270
      WRITE(6, 190 )
      ILINES = 1
  270 WRITE(6, 280 ) J
  280 FORMAT(1H0,40X,37HDETAILED JET HINGE LOADING ON ELEMENT,I4)
      DO 290 M = 1,5
      WRITE(6, 90 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)
  290 CONTINUE
      DO 300 N = 1,NCASES
      LCASE = N
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),
     1           CHORD(K),XB(J),XBB,CPEXP)
  300 CONTINUE
      DO 310 M = 6,10
      WRITE(6, 90 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)
  310 CONTINUE
```

```
      ILINES = ILINES + 12
  320 CONTINUE
      RETURN
      END
      SUBROUTINE STG3FS(CLQ,CMQ,CMQMC,CLLP,NEWMAX,NOALFA,LC)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL SPANWISE AND TOTAL
C  LOADING FOR FUNDAMENTAL CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),
     1           BTA(600),EP(600),CPD(600)
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),
     1           CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1      CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2      CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4      CXCPB(10),CXCLB(10)
      COMMON/LOAD4/ CLG0(40),CLMU0(40),CL0(40),CDMU0(40),CDG0(40),
     1      CDI0(40),CS0(40),CT0(40),CMG0(40),CMMU0(40),CMT0(40),CM0(40),
     2          XBCP0(40),XBCL0(40),FACT(10)
      COMMON/LOAD5/ CGAM(40),CGAM0(40),ALFINF(40),ALFIN0(40),DUMB(40)
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1              CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
   10 N = LC
      LCASE = LC
C
C  LOAD THOSE GEOMETRIC PARAMETERS WHICH ARE DIFFERENT FOR EACH CASE
C  INTO THEIR RESPECTIVE DUMMY ARRAYS
      DO 30  K = 1,NROWS
      TWIST(K) = TST(K,LCASE)
   20 TH(K) = THETA(K,LCASE)
      THETS(K) = THS(K,LCASE)
      HO(K) = HL(K,LCASE)
      DUMB(K) = 0.00
   30 CONTINUE
      DO 50  I = 1,NEWMAX
   40 BTA(I) = BETA(I,LCASE)
      EP(I) = EPS(I,LCASE)
      CPD(I) = CP(I,LCASE)
   50 CONTINUE
C
C  COMPUTE SECTIONAL COEFFICIENTS
      ALPHA = 0.00
      IF((LCASE .EQ. 1) .AND. (NOALFA .NE. 0)) ALPHA = 1.00
   60 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1      XB,DEL,BTA,EP,CPD,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2      CDI,CDMU,CDG,CS,CT0,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL VORTICITY OF WING-JET SYSTEM
      CALL SLOADG(CPD,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)
C
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY
   70 CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)
C
C  COMPUTE TOTAL LINEAR COEFFICIENTS
   80 CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HO,XB,XLEAD,BTA,
     1      CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG(N),CCLJ(N),CCL(N),CCMG(N),
     2      CCMJ(N),CCMT(N),CCM(N),CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),
     3      CXCP(N),CXCL(N),CXCPB(N),CXCLB(N),CCJ(N),CLLG(N),CLLJ(N),
     4      CLL(N),ITYPE,IW,NW,NROWS,ISYMM,
     5      CBGR(N),CBGL(N),CBJR(N),CBJL(N),CBR(N),CBL(N),
     6      CPBMR(N),CPBML(N),CL2R(N),CL2L(N))
C
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS
   90 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,DUMB,
     1 CCDG(N),CCDJ(N),CCS(N),CCDI(N),CDITZ(N),DUMB,ALFINF,DUMMY,CCJ(1),
     2      CNJ(N),CNI(N),CCY(N),XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  DEFINE THE CONSTANT STABILITY DERIVATIVES
      CLQ = CCLG(NCASES)
      CMQ = CCMG(NCASES)
      CMQMC = CMGMC(NCASES)
      CLLP = CLLG(NCASES)
C
C  PRINT LIFT AND DRAG COEFFICIENTS
      IF(IPRINT .GT. 1) RETURN
      WRITE(6, 100 ) LCASE
  100 FORMAT(1H1,36X,11(4H****),2H**/37X,23H*  SPANWISE LOADING FOR,
     1        17H FUNDAMENTAL CASE,I3,3H  */37X,11(4H****),2H**)
      WRITE(6, 110 )
  110 FORMAT(1H ,19X,29H........... LIFT ............,5X,
     1        2H..,7(4H....),14H INDUCED DRAG , 8(4H....)/
     2        1X, 7HSECTION, 5X,1HY,8X,3HCLG,7X,4HCLMU,6X,2HCL,5X,
     3        4H  * ,3X,3HCDG,8X,4HCDMU,7X,2HCS,9X,2HCD,9X,3HCMU,
     4        8X,5HGAMMA,6X,5HALFIN)
      WRITE(6, 120 ) (K,Y(K),CLG(K),CLMU(K),CL(K),CDG(K),CDMU(K),
     1                CS(K),CDI(K),CMU(K),CGAM(K),ALFINF(K),K=1,NROWS)
  120 FORMAT(1H ,I4, 4X, 4F10.6, 4H  * ,7F11.7)
```

```
      WRITE(6, 130 ) CCLG(N),CCLJ(N),CCL(N),CCDG(N),CCDJ(N),CCS(N),
     1          CCDI(N),CCJ(N),CDITZ(N)
  130 FORMAT(1H , 8X, 4(10H ---------),4H  * ,7(11H ----------)/
     1          12X,5HTOTAL,2X,3F10.6,4H  * ,5F11.7,11X,F11.7)
C
C  PRINT PITCHING MOMENT AND CENTER OF LIFT DATA
      IF(NROWS .GT. 21) WRITE(6, 140 )
  140 FORMAT(1H1)
      WRITE(6, 150 )
  150 FORMAT(1H0,24X,39H........... PITCHING MOMENT ...........,10X,
     1          20H ... LIFT CENTER .../
     2          6X, 7HSECTION,5X,1HY,8X,3HCMG,7X,4HCMMU,6X,3HCMT,7X,2HCM,
     3          5X,10H   *   *   ,3X,5HXCP/C,5X,5HXCL/C)
      WRITE(6, 160 ) (K,Y(K),CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)
     1,         K=1,NROWS)
  160 FORMAT(1H ,I9,4X,5F10.6,10H   *   *   ,2F10.6)
      WRITE(6, 170 ) CCMG(N),CCMJ(N),CCMT(N),CCM(N),CXCP(N),CXCL(N),
     1          CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),CXCPB(N),CXCLB(N)
  170 FORMAT(1H ,13X, 5(10H ---------),10H   *   *   , 2(10H ---------)/
     1          17X,5HTOTAL,2X, 4F10.6, 10H  (APEX)  ,2F10.6,9H (X/CREF)/
     2          24X,4F10.6,10H  (XMC)   ,2F10.6,8H (X/B/2))
      RETURN
      END
      SUBROUTINE STG3FT
C
C  THIS SUBROUTINE PRINTS A TABLE OF ALL TOTAL COEFFICIENTS
C  FOR ALL FUNDAMENTAL C SES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1      CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2      CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4      CXCPB(10),CXCLB(10)
      CHARACTER*8 COEFF(35)
      COMMON/LOAD6/COEFF
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1          CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
C  INITIALIZE ALL UNUSED TOTAL COEFFICIENTS TO ZERO FOR PRINTING
      IF(NCASES .EQ. 10) GO TO 20
      NC1 = NCASES + 1
      DO 10  N = NC1,10
        CCLG(N) = 0.00
        CCLJ(N) = 0.00
         CCL(N) = 0.00
        CCDG(N) = 0.00
        CCDJ(N) = 0.00
         CCS(N) = 0.00
        CCDI(N) = 0.00
       CDITZ(N) = 0.00
         CCJ(N) = 0.00
        CCMG(N) = 0.00
        CCMJ(N) = 0.00
        CCMT(N) = 0.00
         CCM(N) = 0.00
        CXCP(N) = 0.00
        CXCL(N) = 0.00
       CXCPB(N) = 0.00
       CXCLB(N) = 0.00
       CMGMC(N) = 0.00
       CMJMC(N) = 0.00
       CMTMC(N) = 0.00
        CMMC(N) = 0.00
        CLLG(N) = 0.00
        CLLJ(N) = 0.00
         CLL(N) = 0.00
         CNJ(N) = 0.00
         CNI(N) = 0.00
         CCY(N) = 0.00
        CBGR(N) = 0.00
        CBGL(N) = 0.00
        CBJR(N) = 0.00
        CBJL(N) = 0.00
         CBR(N) = 0.00
         CBL(N) = 0.00
       CPBMR(N) = 0.00
       CPBML(N) = 0.00
   10 CONTINUE
   20 WRITE(6, 30 ) (N,N=1,10)
   30 FORMAT(1H1,41X, 9(4H****)/
     1       42X,36H*  TOTAL AERODYNAMIC COEFFICIENTS  */42X,9(4H****)//
     2       14X,9(4HCASE,I2,5X), 4HCASE,I3)
   40 FORMAT(1H0, 2X,A8,10F11.7,34(/3X,A8,10F11.7))
      WRITE(6, 40 )COEFF( 1), CCLG, COEFF( 2), CCLJ, COEFF( 3),  CCL,
     1             COEFF( 4), CCDG, COEFF( 5), CCDJ, COEFF( 6),  CCS,
     2             COEFF( 7), CCDI, COEFF( 8),CDITZ, COEFF( 9),  CCJ,
     3             COEFF(10), CCMG, COEFF(11), CCMJ, COEFF(12), CCMT,
     4             COEFF(13),  CCM, COEFF(14), CXCP, COEFF(15), CXCL,
     5             COEFF(16),CXCPB, COEFF(17),CXCLB, COEFF(18),CMGMC,
     6             COEFF(19),CMJMC, COEFF(20),CMTMC, COEFF(21), CMMC,
     7             COEFF(22), CLLG, COEFF(23), CLLJ, COEFF(24),  CLL,
     8             COEFF(25),  CNJ, COEFF(26),  CNI,  COEFF(27), CCY,
     9       COEFF(28),CBGR,COEFF(29),CBGL,COEFF(30),CBJR,
     0       COEFF(31),CBJL,COEFF(32), CBR,COEFF(33),CBL,
     1       COEFF(34),CPBMR,COEFF(35),CPBML
C
   50 RETURN
      END
      SUBROUTINE STG3C(NEWMAX,M,NOALFA)
C
```

```
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE AND
C  TOTAL LOADING FOR THE REQUIRED COMPOSITE CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/CP(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),
     1         BTA(600),EP(600),CP0(600)
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),
     1         CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),
     1     CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),
     2     CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),
     3     CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNI(10),CCY(10),
     4     CXCPB(10),CXCLB(10)
      COMMON/LOAD4/ CLG0(40),CLMU0(40),CL0(40),CDMU0(40),CDG0(40),
     1     CDI0(40),CS0(40),CT0(40),CMG0(40),CMMU0(40),CMT0(40),CM0(40),
     2          XBCP0(40),XBCL0(40),FACT(10)
      COMMON/LOAD5/ CGAM(40),CGAM0(40),ALFINF(40),ALFIN0(40),DUMB(40)
      CHARACTER*8 COEFF(35)
      COMMON/LOAD6/COEFF
      COMMON/LOAD7/CBGR(10),CBGL(10),CBJR(10),CBJL(10),
     1            CBR(10),CBL(10),CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
      DIMENSION CDMUX(40),CDGX(40),CDIX(40),CSX(40),CPA(600)
      EQUIVALENCE (CP(1,1),CPA(1))
C
C  CALCULATE AND PRINT THE CHORDWISE LOADING FOR ALL COMPOSITE CASES
C
      I = 0
      II = NWT
      ILINES = 6
      WRITE(6, 10 ) M
   10 FORMAT(1H1,36X,11(4H****),1H*/
     1            37X,39H*  CHORDWISE LOADING FOR COMPOSITE CASE,I3,3H  */
     2            37X,11(4H****),1H*)
      WRITE(6, 20 ) (N,N=1,10),(FACTOR(N,M),N=1,10)
   20 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),
     1       3X),5X,2HA(,-2,1H)/10X,10F12.6)
      WRITE(6,25)
   25 FORMAT(1H0,7X,47H*** NOTE ***  EACH LEADING EDGE CP VALUE IS THE,
     1        43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION)
      IF(IHINGE .NE. 0) WRITE(6,26)
   26 FORMAT(1H ,21X,47HIF A HINGE IS DEFLECTED THE LOADING IS SINGULAR,
     1     58H AND THE CP(A=0) VALUE IS FOR THE REGULAR EVD PORTION ONLY)
      WRITE(6,27)
   27 FORMAT(1H ,21X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)
      DO 190 K = 1,NROWS
C
C  ON THE WING
      NWK = NW(K)
      DO 50   L = 1,NWK
      I = I + 1
   30 CP0(I) = 0.00
      DO 40  N = 1,NCASES
      FACT(N) = FACTOR(N,M)
      CP0(I) = CP0(I) + CP(I,N) * FACT(N)
   40 CONTINUE
   50 CONTINUE
      NC1 = NCASES + 1
      IF(NC1 .GT. 10) GO TO 70
      DO 60  N = NC1,10
      FACT(N) = 0.00
   60 CONTINUE
   70 J1 = IW(K)
      J2 = IW(K) + NWK - 1
      NEXT = 3 + (2+NOALFA) * (NWK/10+1)
      IF(CMU(K) .GT. 0.0) NEXT = NEXT + (2+NOALFA) * (NJ(K)/10+1) + 1
      ILINES = ILINES + NEXT
      IF((ILINES .LT. 56) .OR. (K .EQ. 1)) GO TO 90
      WRITE(6, 80 )
   80 FORMAT(1H1)
      ILINES = 1
   90 WRITE(6, 100 ) K,Y(K),CHORD(K)
  100 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/
     1         2X,4HWING)
      WRITE(6, 110 ) (XB(J),J=J1,J2)
  110 FORMAT(1H ,7X,2HXB,10F12.6,3(/10X,10F12.6))
      WRITE(6, 120 ) (CP0(J),J=J1,J2)
  120 FORMAT(1H ,2X, 7HCP(A=0),10F12.6,3(/10X,10F12.6))
      IF(NOALFA .GT. 0) WRITE(6, 130 ) (CPA(J),J=J1,J2)
  130 FORMAT(1H ,2X, 7HCP(A=1),10F12.6,3(/10X,10F12.6))
C
C  ON THE JET
      NJK = NJ(K)
      IF(CMU(K) .LT. 0.0001) GO TO 190
      DO 160 L = 1,NJK
      II = II + 1
  140 CP0(II) = 0.00
      DO 150 N = 1,NCASES
      CP0(II) = CP0(II) + CP(II,N) * FACT(N)
  150 CONTINUE
  160 CONTINUE
```

```
  170 J1 = IJ(K)
      J2 = IJ(K) + NJK - 1
      WRITE(6, 180 )
  180 FORMAT(1H ,4H JET)
      WRITE(6, 110 ) (XB(J),J=J1,J2)
      WRITE(6, 120 ) (CP0(J),J=J1,J2)
      IF(NOALFA .GT. 0) WRITE(6, 130 ) (CPA(J),J=J1,J2)
  190 CONTINUE
C
C   COMPUTE AND PRINT SPANWISE AND TOTAL LOADINGS FOR EACH COMPOSITE CASE
C
C   DEFINE THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 1
      IF(NOALFA .EQ. 0) GO TO 250
      DO 210 K = 1,NROWS
  200 TWIST(K) = TST(K,1)
      TH(K) = THETA(K,1)
      THETS(K) = THS(K,1)
  210 CONTINUE
      DO 230 I = 1,NEWMAX
  220 BTA(I) = BETA(I,1)
      EP(I) = EPS(I,1)
  230 CONTINUE
C
C   COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 1
      ALPHA = 1.0
  240 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1         XB,DEL,BTA,EP,CPA,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2         CDI,CDMU,CDG,CS,CT0,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL VORTICITY FOR ALPHA = 1
      CALL SLOADG(CPA,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 1
      CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)
C
C   MODULATE AND SUM THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 0
  250 DO 280 K = 1,NROWS
      TWIST(K) = 0.00
      TH(K) = 0.00
      THETS(K) = 0.00
      H0(K) = 0.00
      DO 270 N = 1,NCASES
  260 TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)
      TH(K) = TH(K) + THETA(K,N) * FACT(N)
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)
      H0(K) = H0(K) + HL(K,N) * FACT(N)
  270 CONTINUE
  280 CONTINUE
      DO 320 I = 1,NEWMAX
  290 BTA(I) = 0.00
      EP(I) = 0.00
      DO 310 N = 1,NCASES
  300 BTA(I) = BTA(I) + BETA(I,N) * FACT(N)
      EP(I) = EP(I) + EPS(I,N) * FACT(N)
  310 CONTINUE
  320 CONTINUE
C
C   COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 0
      ALPHA = 0.00
      IF(NOALFA .GT. 0) ALPHA = FACT(1)
  330 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1         XB,DEL,BTA,EP,CP0,CL0,CLG0,CLMU0,CM0,CMG0,CMMU0,CMT0,
     2         XBCP0,XBCL0,CDI0,CDMU0,CDG0,CS0,CT0,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL VORTICITY FOR ALPHA = 0
  340 CALL SLOADG(CP0,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG0,CGAM0,NROWS,IHINGE)
C
C   COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 0
  350 CALL TREFTZ(Y,DELTA,CMU,CGAM0,ALFIN0,NROWS,ISYMM)
C
C   COMPUTE SECTIONAL NONLINEAR CROSS-PRODUCT COEFFICIENTS
      IF(NOALFA .EQ. 0) GO TO 370
  360 CALL SLOADX(CPA,CP0,DEL,EP,CMU,TH,NW,NJ,IJ,
     1               CLG0,CDGX,CDMUX,CSX,CDIX,NROWS)
C
C   COMPUTE TOTAL LINEAR COEFFICIENTS FOR ALPHA = 0
  370 CALL TLOAD0(CREF,CCLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,
     1         CLLG,CLLJ,FACT,CCLG0,CCLJ0,CCL0,CCMG0,CCMJ0,CCMT0,CCM0,
     2         CMGMC0,CMJMC0,CMTMC0,CMMC0,CXCP0,CXCL0,CXCPB0,CXCLB0,
     3         CLLG0,CLLJ0,CLL0,NCASES,ISYMM,
     4         CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L,
     5         CBGR0,CBGL0,CBJR0,CBJL0,CBR0,CBL0,CPBMR0,CPBML0,CL2R0,CL2L0)
C
C   COMPUTE TOTAL NONLINEAR COEFFICIENTS FOR ALPHA = 0
  380 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG0,CDMU0,CS0,CDI0,CL0,DUMB,
     1           CCDG0,CCDJ0,CCS0,CCDI0,CDITZ0,DUMB,ALFIN0,CCT0,CCJ(1),
     2           DUMMY,CNI0,CCY0,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C   COMPUTE TOTAL NONLINEAR CROSS-PRODUCT COEFFICIENTS
      IF(NOALFA .EQ. 0) GO TO 400
  390 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGX,CDMUX,CSX,CDIX,CL,CL0,
     1           CCDGX,CCDJX,CCSX,CCDIX,CDITZX,ALFINF,ALFIN0,DUMMY,CCJ(1),
     2           DUMMY,CNIX,CCYX,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C   PRINT THE SECTIONAL AND TOTAL COEFFICIENTS
C
  400 WRITE(6, 410 ) M
  410 FORMAT(1H1,47X,6(4H****)/48X,18H*  COMPOSITE  CASE,I3,3H  */
     1              48X,6(4H****))
      WRITE(6, 420 ) (N,N=1,10),(FACT(N),N=1,10)
  420 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(4X,2HA(,I1,1H),
```

```
     1         2X),3X,2HA(,I2,1H)/10X,10F10.6/)
      WRITE(6, 430 )
  430 FORMAT(1H ,20X,29H............ LIFT ............,11H   *  *    ,
     1         39H........... PITCHING MOMENT ..............,10H   *  *   ,
     2         20H ... LIFT CENTER .../ 2X,7HSECTION,5X,1HY,8X,4HCLG0,6X,
     3         5HCLMU0,5X,3HCL0,4X,10H   *  *   ,3X,4HCMG0,6X,5HCMMU0,5X,
     4         4HCMT0,6X,3HCM0,4X,10H   *  *   ,3X,6HXCP0/C,4X,6HXCL0/C)
      IF(NOALFA .GT. 0) WRITE(6, 440 )
  440 FORMAT(1H ,
     1         22X,4HCLGA,6X,5HCLMUA,5X,3HCLA,4X,10H   *  *   ,3X,4HCMGA,
     2         6X,5HCMMUA,5X,4HCMTA,6X,3HCMA,4X,10H   *  *   ,3X,6HXCPA/C,
     3         4X,6HXCLA/C)
      DO 470 K = 1,NROWS
      WRITE(6, 450 ) K,Y(K),CLG0(K),CLMU0(K),CL0(K),CMG0(K),CMMU0(K),
     1               CMT0(K),CM0(K),XBCP0(K),XBCL0(K)
  450 FORMAT(1H ,I5,4X,4F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 460 ) CLG(K),CLMU(K),CL(K),
     1               CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)
  460 FORMAT(1H ,19X,3F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)
  470 CONTINUE
      WRITE(6, 480 ) CCLG0,CCLJ0,CCL0,CCMG0,CCMJ0,CCMT0,CCM0,CXCP0,CXCL0
  480 FORMAT(1H , 9X,4(10H ---------),10H   *  *   ,4(10H ---------),
     1         10H   *  *   ,2(10H ---------)/13X,5HTOTAL,2X,3F10.6,
     2         10H   *  *   ,4F10.6,10H  (APEX)  , 2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 490 ) CCLG(1),CCLJ(1),CCL(1),
     1               CCMG(1),CCMJ(1),CCMT(1),CCM(1),CXCP(1),CXCL(1)
  490 FORMAT(1H ,19X,3F10.6,
     1         10H   *  *   ,4F10.6,10H  (APEX)  , 2F10.6)
      WRITE(6, 500 ) CMGMC0,CMJMC0,CMTMC0,CMMC0,CXCPB0,CXCLB0
  500 FORMAT(1H ,49X,
     1         10H   *  *   ,4F10.6,10H  (XMC)   , 2F10.6)
      IF(NOALFA .GT. 0) WRITE(6, 510 )
     1               CMGMC(1),CMJMC(1),CMTMC(1),CMMC(1),CXCPB(1),CXCLB(1)
  510 FORMAT(1H ,49X,10H   *  *   ,4F10.6,10H  (XMC)   , 2F10.6)
      WRITE(6, 520 )
  520 FORMAT(1H1,25X,9(4H....),15H INDUCED DRAG .,9(4H....)/
     1      7X,7HSECTION,6X,1HY,7X,4HCDG0,7X,5HCDMU0,6X,3HCS0,8X,3HCD0,
     2    8X,6HGAMMA0,5X,6HALFIN0,5X,3HCT0,8X,3HCMU)
      IF(NOALFA .GT. 0) WRITE(6, 530 )
  530 FORMAT(1H ,27X,4HCDGX,7X,5HCDMUX,6X,3HCSX,8X,3HCDX/
     1    28X,5HCDGA2,6X,6HCDMUA2,5X,4HCSA2,7X,4HCDA2,7X,6HGAMMA2,
     2    5X,6HALFINA)
      DO 560 K = 1,NROWS
      WRITE(6, 540 ) K,Y(K),CDG0(K),CDMU0(K),CS0(K),CDI0(K),
     1       CGAM0(K),ALFIN0(K),CT0(K),CMU(K)
  540 FORMAT(1H , 6X,I4 ,4X,F10.6,8F11.7)
      IF(NOALFA .GT. 0) WRITE(6, 550 )
     1       CDGX(K),CDMUX(K),CSX(K),CDIX(K),CDG(K),CDMU(K),CS(K),
     2       CDI(K),CGAM(K),ALFINF(K)
  550 FORMAT(1H ,24X,4F11.7/25X,6F11.7)
  560 CONTINUE
      WRITE(6, 570 ) CCDG0,CCDJ0,CCS0,CCDI0,CDITZ0,CCT0,CCJ(1)
  570 FORMAT(1H ,24X,8(11H ----------)/18X,5HTOTAL,2X,4F11.7,11X,3F11.7)
      IF(NOALFA .GT. 0) WRITE(6, 580 ) CCDGX,
     1    CCDJX,CCSX,CCDIX,CDITZX,CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1)
  580 FORMAT(1H ,24X,4F11.7,11X,F11.7/25X,4F11.7,11X,F11.7)
C
C  PRINT A TABLE OF ALL TOTAL COEFFICIENTS FOR ALPHA = 0,ALPHA,ALPHA**2
C
      WRITE(6, 590 )
  590 FORMAT(1H1,41X,9(4H****)/42X,20H*  TOTAL AERODYNAMIC,
     1         16H COEFFICIENTS  */42X,9(4H****)//
     2         42X, 7HALPHA=0,   8X, 5HALPHA,10X, 8HALPHA**2)
      IF(NOALFA .GT. 0) GO TO 630
  600 FORMAT(1H ,25X,A8,F15.7)
      WRITE(6, 600 )COEFF( 1),CCLG0,COEFF( 2), CCLJ0,COEFF(3),CCL0,
     1                 COEFF( 4),CCDG0,COEFF( 5), CCDJ0,COEFF(6),CCS0,
     2                 COEFF( 7),CCDI0,COEFF( 8),CDITZ0,COEFF(9),CCJ(1),
     3                 COEFF(10),CCMG0,COEFF(11), CCMJ0,COEFF(12),CCMT0,
     4                 COEFF(13), CCM0,COEFF(14), CXCP0,COEFF(15),CXCL0,
     5                 COEFF(16),CXCPB,COEFF(17), CXCLB,COEFF(18),CMGMC0,
     6                COEFF(19),CMJMC0,COEFF(20),CMTMC0,COEFF(21),CMMC0,
     7                COEFF(22), CLLG0,COEFF(23), CLLJ0,COEFF(24),CLL0,
     8                COEFF(25),CNJ(1),COEFF(26),  CNI0,COEFF(27),CCY0,
     9                COEFF(28), CBGR0,COEFF(29), CBGL0,COEFF(30),CBJR0,
     0                COEFF(31), CBJL0,COEFF(32),  CBR0,COEFF(33),CBL0,
     1                COEFF(34),CPBMR0,COEFF(35),CPBML0
      RETURN
C
  610 FORMAT(1H ,25X,A8,3F15.6)
  620 FORMAT(1H ,25X,A8,1X,3F15.7)
  630 WRITE(6, 610 )COEFF( 1), CCLG0,CCLG(1)
      WRITE(6, 610 )COEFF( 2), CCLJ0,CCLJ(1)
      WRITE(6, 610 )COEFF( 3), CCL0 ,CCL(1)
      WRITE(6, 620 )COEFF( 4), CCDG0,CCDGX,CCDG(1)
      WRITE(6, 620 )COEFF( 5), CCDJ0,CCDJX,CCDJ(1)
      WRITE(6, 620 )COEFF( 6), CCS0 ,CCSX ,CCS(1)
      WRITE(6, 620 )COEFF( 7), CCDI0,CCDIX,CCDI(1)
      WRITE(6, 620 )COEFF( 8), CDITZ0,CDITZX,CDITZ(1)
      WRITE(6, 620 )COEFF( 9), CCJ(1)
      WRITE(6, 610 )COEFF(10), CCMG0,CCMG(1)
      WRITE(6, 610 )COEFF(11), CCMJ0,CCMJ(1)
      WRITE(6, 610 )COEFF(12), CCMT0,CCMT(1)
      WRITE(6, 610 )COEFF(13), CCM0 ,CCM(1)
      WRITE(6, 610 )COEFF(14), CXCP0,CXCP(1)
      WRITE(6, 610 )COEFF(15), CXCL0,CXCL(1)
      WRITE(6, 610 )COEFF(16), CXCPB0,CXCPB(1)
      WRITE(6, 610 )COEFF(17), CXCLB0,CXCLB(1)
      WRITE(6, 610 )COEFF(18), CMGMC0,CMGMC(1)
      WRITE(6, 610 )COEFF(19), CMJMC0,CMJMC(1)
      WRITE(6, 610 )COEFF(20), CMTMC0,CMTMC(1)
```

```
      WRITE(6, 610 )COEFF(21), CMMCO,CMMC(1)
      WRITE(6, 610 )COEFF(22), CLLGO,CLLG(1)
      WRITE(6, 610 )COEFF(23), CLLJO,CLLJ(1)
      WRITE(6, 610 )COEFF(24), CLLO,CLL(1)
      WRITE(6, 610 )COEFF(25), CNJ(1)
      WRITE(6, 610 )COEFF(26), CNIO,CNIX,CNI(1)
      WRITE(6, 610 )COEFF(27), CCYO,CCYX,CCY(1)
      WRITE(6, 610 )COEFF(28), CBGRO,  CBGR(1)
      WRITE(6, 610 )COEFF(29), CBGLO,  CBGL(1)
      WRITE(6, 610 )COEFF(30), CBJRO,  CBJR(1)
      WRITE(6, 610 )COEFF(31), CBJLO,  CBJL(1)
      WRITE(6, 610 )COEFF(32), CBRO,   CBR(1)
      WRITE(6, 610 )COEFF(33), CBLO,   CBL(1)
      WRITE(6, 610 )COEFF(34), CPBMRO, CPBMR(1)
      WRITE(6, 610 )COEFF(35), CPBMLO, CPBML(1)
C
C  COMPUTE AND PRINT A TABLE OF THE VARIATION OF THE TOTAL COEFFICIENTS
C  WITH ANGLE OF ATTACK
  640 CALL TABLE(CCLO,CCL(1),CMMCO,CMMC(1),CLLO,CLL(1),CDITZO,CDITZX,
     1             CDITZ(1),CCJ(1),CNIO,CNIX,CNI(1),CNJ(1),
     2             CCYO,CCYX,CCY(1),M)
C
      RETURN
      END
      BLOCK DATA
      CHARACTER*8 COEFF(35)
      COMMON/LOAD6/COEFF
      DATA COEFF/'    CCLG','    CCLJ','**   CCL','    CCDG','    CCDJ',
     1           '     CCS','     CCD','** CDITZ','**   CCJ','    CCMG',
     2           '    CCMJ','    CCMT','     CCM','    CXCP','    CXCL',
     3           '   CXCPB','   CXCLB','  CCMGMC','  CCMJMC','  CCMTMC',
     4           '** CCMMC','    CLLG','    CLLJ','*    CLL','     CNJ',
     5           '*  CNIMC','*    CCY',
     6           '    CBGR','    CBGL','    CBJR','    CBJL',
     7           '     CBR','     CBL','   CPMBR','   CPMBL'/
      END
      SUBROUTINE EXPLE(LCASE,CPI,CPI1,DEL,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF A LEADING EDGE EVD
C  AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
   10 DN = 0.20
      DO 40 N = 1,5
   20 X = DN * N
      XBB(N) = X * DEL
   30 CPEXP(N,LCASE) = 0.666666*CPI*(1.0/SQRT(X)-X) + CPI1*X
   40 CONTINUE
      RETURN
      END
      SUBROUTINE EXPH1(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE FORWARD HALF OF A HINGE
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
      DN = 0.20
      DO 30  N = 1,5
   10 X = XB - DEL + (N-1)*DN*DEL
      DX = X - XB
      XBB(N) = X
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(-C*DX)
     1      + ALOG(C*DEL)*DX/DEL) +(CPI + (CPI-CPI1)*DX/DEL)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE EXPH2(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)
C
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE REAR HALF OF A HINGE
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT
C
      DIMENSION XBB(5),CPEXP(5,10)
C
      DN = 0.20
      DO 30  N = 1,5
   10 DX = N*DN*DEL
      X = XB + DX
      XBB(N) = X
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(C*DX)
     1      - ALOG(C*DEL)*DX/DEL) + (CPI - (CPI-CPI1)*DX/DEL)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,
     1    XB,DEL,BETA,EPS,CP,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,
     2    CDI,CDMU,CDG,CS,CT,NROWS,IHINGE)
C
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF LIFT, PITCHING
C  MOMENT, AND INDUCED DRAG FOR EITHER A FUNDAMENTAL OR A COMPOSITE CASE
C
      DIMENSION IJ(40),NW(40),NJ(40)
      DIMENSION CHORD(40),CMU(40),THETA(40),THETAS(40),TST(40)
      DIMENSION CP(600),XB(600),DEL(600),BETA(600),EPS(600)
      DIMENSION CLG(40),CLMU(40),CL(40),CMG(40),CMMU(40),CMT(40),CM(40),
     1    XBCP(40),XBCL(40),CDG(40),CDMU(40),CS(40),CDI(40),CT(40)
C
C  INTEGRATE THE CHORDWISE PRESSURES FOR EACH SPANWISE SECTION
   10 I = 0
      IF(IHINGE .GT. 1) IHINGE = 1
```

```
      DO 150 K = 1,NROWS
C
C  LEADING EDGE CONTRIBUTIONS
      I = I + 1
   20 CLI = DEL(I) * (CP(I)+0.50*CP(I+1))
      CLG(K) = CLI
      CMG(K) = -DEL(I)**2 * (0.6666667*CP(I)+CP(I+1)) / 3.00
      CDG(K) = CLI * EPS(I)/57.295779
   30 CS(K) = 0.1745329 * DEL(I) * CP(I)**2
C
      BCF = 0.00
      NWK = NW(K)
      DO 100 L = 2,NWK
      I = I + 1
      CPI = CP(I)
      IF(L .EQ. NWK) GO TO 40
      CPI1 = CP(I+1)
      GO TO 50
C  DEFINE TRAILING EDGE CP VALUE
   40 CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 50
      IJK = IJ(K)
      CPI1 = CP(IJK)
C
C  REGULAR EVD CONTRIBUTIONS
   50 CLI = 0.50 * DEL(I) * (CPI+CPI1)
      CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.00
   60 CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
      BCF = BCF + BETA(I) * (1.0-XB(I))
C
C  HINGE CONTRIBUTIONS
      IF(IHINGE .EQ. 0) GO TO 100
      B2 = BETA(I+1)
      IF(L .LT. NWK) GO TO 70
      B2 = 0.00
      IJK = IJ(K)
      IF(CMU(K) .GT. 0.0001) B2 = BETA(IJK)
   70 IF((ABS(BETA(I)) .LT. 0.0001).AND.(ABS(B2) .LT. 0.0001)) GO TO 100
      CLI = 0.00
      CMI = 0.00
      DL = ALOG(DEL(I) * CHORD(K))
      CON = 0.6366198 * DEL(I) / 57.295779
      IF(ABS(BETA(I)) .LT. 0.0001) GO TO 80
      CLI = CON * BETA(I) * (2.00 - DL)
      CMI = BETA(I) * (0.50-DL/3.00)
   80 IF(ABS(B2) .LT. 0.0001) GO TO 90
      CLI = CLI + CON * B2 * (2.00 - DL)
      CMI = CMI + 2.00 * B2 * (0.7500 - DL/3.00)
   90 CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CON * DEL(I) * CMI - CLI * XB(I)
      CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
  100 CONTINUE
C
C  COMPUTE THE SECTIONAL COEFFICIENTS
  110 CLMU(K) = CMU(K) * THETA(K)/57.295779
      CL(K) = CLG(K) + CLMU(K)
  120 CMMU(K) = -CMU(K) * THETA(K)/57.295779
      CMT(K) = CMU(K) * (ALPHA+TST(K)-THETAS(K)+BCF)/57.295779
      CM(K) = CMG(K) + CMMU(K) + CMT(K)
      XBCP(K) = 0.00
      XBCL(K) = 0.00
  130 IF(CLG(K) .NE. 0.00) XBCP(K) = -CMG(K) / CLG(K)
      IF(CL(K) .NE. 0.00) XBCL(K) = -(CMG(K)+CMMU(K)) / CL(K)
  140 CDMU(K) = CMU(K) * (THETA(K)/57.295779)**2 / 2.00
      CDI(K) = CDG(K) + CDMU(K) - CS(K)
      CT(K) = CMU(K) - CDI(K)
  150 CONTINUE
  160 RETURN
      END
      SUBROUTINE SLOADX(CPA,CP0,DEL,EPS,CMU,TH,NW,NJ,IJ,
     1                  CLG0,CDGX,CDMUX,CSX,CDIX,NROWS)
C
C  THIS SUBROUTINE CALCULATES THE SECTIONAL CROSS-PRODUCT VALUES
C  OF THE NONLINEAR DRAG COEFFICIENTS
C
      DIMENSION CPA(600),CP0(600),DEL(600),EPS(600)
      DIMENSION CMU(40),TH(40),NW(40),NJ(40),IJ(40)
      DIMENSION CLG0(40),CDGX(40),CDMUX(40),CSX(40),CDIX(40)
C
      I = 0
      DO 70  K = 1,NROWS
C  LEADING EDGE EVD CONTRIBUTION
      I = I + 1
   10 CDGX(K) = DEL(I)*(CPA(I)+0.50*CPA(I+1))*EPS(I)/57.295779
   20 CSX(K) = 0.3490658 * DEL(I) * (CP0(I) * CPA(I))
      NWK = NW(K)
      DO 50 L = 2,NWK
      I = I + 1
   30 CPI1 = CPA(I+1)
C  DEFINE TRAILING EDGE CP VALUE
      IF(L .LT. NWK) GO TO 40
      CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 40
      IJK = IJ(K)
      CPI1 = CPA(IJK)
C  REGULAR EVD CONTRIBUTION
   40 CDGX(K) = CDGX(K) + 0.50*DEL(I)*(CPA(I)+CPI1)*EPS(I)/57.295779
   50 CONTINUE
C
C  COMPUTE THE REMAINING SECTIONAL COEFFICIENTS
```

```
   60 CDGX(K) = CDGX(K) + CLGO(K)/57.295779
      CDMUX(K) = CMU(K) * TH(K)/57.295779**2
      CDIX(K) = CDGX(K) + CDMUX(K) - CSX(K)
   70 CONTINUE
      RETURN
      END
      SUBROUTINE SLOADG(CP,DEL,BETA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,
     1                  NROWS,IHINGE)
C
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF TOTAL VORTICITY
C  ON THE WING-JET SYSTEM
C
      DIMENSION CP(600),DEL(600),BETA(600)
      DIMENSION CHORD(40),D(40),CMU(40),NJ(40),IJ(40)
      DIMENSION CLG(40),CGAM(40)
C
      DO 60  K = 1,NROWS
C
C  COMPUTE THE SECTIONAL JET VORTICITY, INTEDRATED FROM T.E. TO INFINITY
   10 CGAM(K) = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      II = IJ(K)
C  HINGE EVD CONTRIBUTION
      IF(IHINGE .EQ. 0) GO TO 20
      IF(BETA(II) .NE. 0.00) CGAM(K) = 0.6366198 * DEL(II) *
     1      BETA(II)/57.295779 * (2.00-ALOG(CHORD(K)*DEL(II)))
C  REGULAR EVD CONTRIBUTION
   20 NJK1 = NJ(K) - 1
      II = II - 1
      DO 40 L = 1,NJK1
      II = II + 1
   30 CGAM(K) = CGAM(K) + 0.50 * DEL(II) * (CP(II)+CP(II+1))
   40 CONTINUE
C  FAR-JET EVD CONTRIBUTION
      II = II + 1
      CGAM(K) = CGAM(K) +        D(K) / CHORD(K) * CP(II)
C
C  SUM UP THE WING AND JET CONTRIBUTIONS
   50 CGAM(K) = 0.50 * CHORD(K) * (CGAM(K)+CLG(K))
   60 CONTINUE
      RETURN
      END
      SUBROUTINE TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,C,HO,XB,XLEAD,BETA,
     1     CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,
     2     CMGMC,CMJMC,CMTMC,CMMC,CXCP,CXCL,CXCPB,CXCLB,CCJ,CLLG,CLLJ,
     3     CLL,ITYPE,IW,NW,NROWS,ISYMM,
     4     CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L)
C
C  THIS SUBROUTINE CALCULATES ALL OF THE TOTAL LOADING PARAMETERS
C  FOR A FUNDAMENTAL CASE
C
      DIMENSION Y(40),DELTA(40),C(40),HO(40),XLEAD(40),IW(40),NW(40)
      DIMENSION XB(600),BETA(600),ITYPE(600)
      DIMENSION CLG(40),CLMU(40),CMG(40),CMMU(40),CMT(40),CMU(40)
C
C  INITIALIZE THE TOTAL COEFFICIENTS
   10 CCLG = 0.00
      CCLJ = 0.00
      CCMG = 0.00
      CCMJ = 0.00
      CCMT = 0.00
      CXCP = 0.00
      CXCL = 0.00
      CXCPB = 0.00
      CXCLB = 0.00
      CCJ  = 0.00
      CLLG = 0.00
      CLLJ = 0.00
      CLL  = 0.00
      CBGR = 0.00
      CBGL = 0.00
      CBJR = 0.00
      CBJL = 0.00
      CBR = 0.00
      CBL = 0.00
      CL2R = 0.00
      CL2L = 0.00
      CPBMR = 0.00
      CPBML = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 100 K = 1,NROWS
   20 CDEL = C(K) * DELTA(K)
      IF(ISYMM .LT. 0) GO TO 80
C
C  LIFT COEFFICIENTS
      CCLG = CCLG + CDEL * CLG(K)
   30 CCLJ = CCLJ + CDEL * CLMU(K)
C
C  PITCHING MOMENT COEFFICIENTS
      CCDEL = CDEL * C(K)
      XLB = XLEAD(K) / C(K)
C  COMPUTE LEADING EDGE HEIGHT ABOVE WING APEX
   40 I = IW(K) - 1
      NWK = NW(K)
      XDS = 0.00
      DO 60 L = 1,NWK
      I = I + 1
      IF(ITYPE(I) - 41) 60 , 50 , 70
   50 XDS = XDS + XB(I) * BETA(I)/57.295779
   60 CONTINUE
```

```
   70 HLB = HO(K) - XLB * ALPHA/57.295779 - XDS
      CCMG = CCMG + CCDEL * (CMG(K) - CLG(K)*XLB)
      CCMJ = CCMJ + CCDEL * (CMMU(K) - CLMU(K)*XLB)
      CCMT = CCMT + CCDEL * (CMT(K) - CMU(K)*HLB)
   80 CCJ  = CCJ  + CDEL * CMU(K)
C
C  ROLLING MOMENT COEFFICIENTS AND ROOT BENDING MOMENTS
      CDELY = CDEL * Y(K)
   90 IF(ISYMM .EQ. 0) GO TO 95
      CLLG = CLLG + CDELY * CLG(K)
      CLLJ = CLLJ + CDELY * CLMU(K)
   95 IF(Y(K) .LT. 0.0) GO TO 96
      CBGR = CBGR + CDEL  * CLG(K)
      CBJR = CBJR + CDEL  * CLMU(K)
      CL2R = CL2R + CDEL * (CLG(K)+CLMU(K))
      GO TO 100
   96 CBGL = CBGL - CDELY * CLG(K)
      CBJL = CBJL - CDELY * CLMU(K)
      CL2L = CL2L + CDEL * (CLG(K)+CLMU(K))
  100 CONTINUE
C
C  COMPUTE THE FINAL VALUES OF ALL THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
  110 CCLG = FACTOR * CCLG
      CCLJ = FACTOR * CCLJ
      CCL  = CCLG + CCLJ
  120 CCJ  = FACTOR * CCJ
  130 FACTOR = FACTOR / CREF
      CCMG = FACTOR * CCMG
      CCMJ = FACTOR * CCMJ
      CCMT = FACTOR * CCMT
      CCM  = CCMG + CCMJ + CCMT
      IF(ISYMM .LT. 0) GO TO 140
      IF(CCLG .NE. 0.00) CXCP = - CCMG / CCLG
      IF(CCL .NE. 0.00) CXCL = -(CCMG+CCMJ) / CCL
      CXCPB = CXCP * CREF
      CXCLB = CXCL * CREF
  140 FACTOR = XMC / CREF
      IF(ISYMM .LT. 0) FACTOR = 0.00
      CMGMC = CCMG + CCLG * FACTOR
      CMJMC = CCMJ + CCLJ * FACTOR
      CMTMC = CCMT - CCJ  * FACTOR * ALPHA/57.295779
      CMMC  = CMGMC + CMJMC + CMTMC
  150 IF(ISYMM .EQ. 0) GO TO 160
      FACTOR = -1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = -2.00 / AREA
      CLLG = FACTOR * CLLG
      CLLJ = FACTOR * CLLJ
      CLL  = CLLG + CLLJ
  160 FACTOR = 4.0 / AREA
      CBGR = FACTOR * CBGR
      CBJR = FACTOR * CBJR
      CL2R = FACTOR * CL2R
      IF(ISYMM .GT. 0) GO TO 170
      IF(ISYMM .LT. 0) GO TO 180
      CBGL = CBGR
      CBJL = CBJR
      CL2L = CL2R
      GO TO 190
  170 CBGL = FACTOR * CBGL
      CBJL = FACTOR * CBJL
      CL2L = FACTOR * CL2L
      GO TO 190
  180 CBGL = -CBGR
      CBJL = -CBJR
      CL2L = -CL2R
  190 CBR = CBGR + CBJR
      CBL = CBGL + CBJL
      IF(CL2R .NE. 0.0) CPBMR = CBR/CL2R
      IF(CL2L .NE. 0.0) CPBML = CBL/CL2L
      RETURN
      END
      SUBROUTINE TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,CLO,
     1                  CCDG,CCDJ,CCS,CCDI,CDITZ,ALFINF,ALFINO,CCT,CCJ,
     2                  CNJ,CNI,CCY,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  THIS SUBROUTINE CALCULATES THE NONLINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0 BY SPANWISE INTEGRATION OF THE NONLINEAR
C  SECTIONAL COEFFICIENTS
C
      DIMENSION CHORD(40),DELTA(40),Y(40),CMU(40),XLEAD(40),TANLE(40)
      DIMENSION CDG(40),CDMU(40),CS(40),CDI(40),CL(40),CLO(40)
      DIMENSION ALFINF(40),ALFINO(40)
C
C  INITIALIZE THE COEFFICIENTS
   10 CCDG = 0.00
      CCDJ = 0.00
      CCS = 0.00
      CDITZ = 0.00
      CNJ = 0.00
      CNI = 0.00
      CCY = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 40  K = 1,NROWS
   20 CDEL = CHORD(K) * DELTA(K)
      CCDG = CCDG + CDEL * CDG(K)
      CCDJ = CCDJ + CDEL * CDMU(K)
   30 CCS = CCS + CDEL * CS(K)
      CDITZ = CDITZ + CDEL * (CL(K)*ALFINO(K) + CLO(K)*ALFINF(K))
```

```
      IF(ISYMM .LT. 1) GO TO 40
      CNJ = CNJ + CDEL * Y(K) * CMU(K)
      CNI = CNI + CDEL * (Y(K)*CDI(K) - CS(K)*TANLE(K)*(XLEAD(K)-XMC))
      CCY = CCY + CDEL * CS(K) * TANLE(K)
   40 CONTINUE
C  COMPUTE THE FINAL VALUES OF THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
   50 CCDG = FACTOR * CCDG
      CCDJ = FACTOR * CCDJ
      CCS = FACTOR * CCS
      CDITZ = FACTOR * CDITZ / 2.00
      CCDI = CCDG + CCDJ - CCS
      CCT = CCJ - CCDI
   60 IF(ISYMM .LT. 1) RETURN
      FACTOR = 1.00 / AREA
      CNJ = -FACTOR * CNJ
      CNI = FACTOR * CNI
      CCY = FACTOR * CCY
   70 RETURN
      END
      SUBROUTINE TLOAD0(CREF,CCLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,
     1                  CLLG,CLLJ,FACT,CCLG0,CCLJ0,CCL0,CCMG0,CCMJ0,CCMT0,CCM0,
     2                  CMGMC0,CMJMC0,CMTMC0,CMMC0,CXCP0,CXCL0,CXCPB0,CXCLB0,
     3                  CLLG0,CLLJ0,CLL0,NCASES,ISYMM,
     4          CBGR,CBGL,CBJR,CBJL,CBR,CBL,CPBMR,CPBML,CL2R,CL2L,
     5          CBGR0,CBGL0,CBJR0,CBJL0,CBR0,CBL0,CPBMR0,CPBML0,CL2R0,CL2L0)
C
C  THIS SUBROUTINE CALCULATES THE LINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0.  LINEAR QUANTITIES ARE MODULATED AND SUMMED
C  ACCORDING TO THE COMPOSITE CASE REQUIREMENTS.
C
      DIMENSION CCLG(10),CCLJ(10),CCMG(10),CCMJ(10),CCMT(10),
     1          CMGMC(10),CMJMC(10),CMTMC(10),CLLG(10),CLLJ(10),FACT(10)
      DIMENSION CBGR(10),CBGL(10),CBJR(10),CBJL(10),CBR(10),CBL(10),
     1          CPBMR(10),CPBML(10),CL2R(10),CL2L(10)
C
C  INITIALIZE THE COEFFICIENTS
   10 CCLG0 = 0.00
      CCLJ0 = 0.00
      CCMG0 = 0.00
      CCMJ0 = 0.00
      CCMT0 = 0.00
      CMGMC0 = 0.00
      CMJMC0 = 0.00
      CMTMC0 = 0.00
      CLLG0 = 0.00
      CLLJ0 = 0.00
      CXCP0 = 0.00
      CXCL0 = 0.00
      CXCPB0 = 0.00
      CXCLB0 = 0.00
      CBGR0 = 0.00
      CBGL0 = 0.00
      CBJR0 = 0.00
      CBJL0 = 0.00
      CBR0 = 0.00
      CBL0 = 0.00
      CL2R0 = 0.00
      CL2L0 = 0.00
      CPBMR0 = 0.00
      CPBML0 = 0.00
C
C  MODULATE AND SUM THE TOTAL COEFFICIENTS
      DO 40 N = 1,NCASES
   20 CCLG0 = CCLG0 + CCLG(N) * FACT(N)
      CCLJ0 = CCLJ0 + CCLJ(N) * FACT(N)
      CCMG0 = CCMG0 + CCMG(N) * FACT(N)
      CCMJ0 = CCMJ0 + CCMJ(N) * FACT(N)
      CCMT0 = CCMT0 + CCMT(N) * FACT(N)
   30 CMGMC0 = CMGMC0 + CMGMC(N) * FACT(N)
      CMJMC0 = CMJMC0 + CMJMC(N) * FACT(N)
      CMTMC0 = CMTMC0 + CMTMC(N) * FACT(N)
      CLLG0 = CLLG0 + CLLG(N) * FACT(N)
      CLLJ0 = CLLJ0 + CLLJ(N) * FACT(N)
      CBGR0 = CBGR0 + CBGR(N) * FACT(N)
      CBGL0 = CBGL0 + CBGL(N) * FACT(N)
      CBJR0 = CBJR0 + CBJR(N) * FACT(N)
      CBJL0 = CBJL0 + CBJL(N) * FACT(N)
      CL2R0 = CL2R0 + CL2R(N) * FACT(N)
      CL2L0 = CL2L0 + CL2L(N) * FACT(N)
   40 CONTINUE
C  DEFINE THE REMAINING TOTAL COEFFICIENTS
      CCL0 = CCLG0 + CCLJ0
   50 CCM0 = CCMG0 + CCMJ0 + CCMT0
      CBR0 = CBGR0 + CBJR0
      CBL0 = CBGL0 + CBJL0
      IF(CL2R0 .NE. 0.0) CPBMR0 = CBR0 / CL2R0
      IF(CL2L0 .NE. 0.0) CPBML0 = CBL0 / CL2L0
      IF(ISYMM .LT. 0) GO TO 60
      IF(CCLG0 .NE. 0.00) CXCP0 = - CCMG0 / CCLG0
      IF(CCL0 .NE. 0.00) CXCL0 = - (CCMG0+CCMJ0) / CCL0
      CXCPB0 = CXCP0 * CREF
      CXCLB0 = CXCL0 * CREF
   60 CMMC0 = CMGMC0 + CMJMC0 + CMTMC0
      CLL0 = CLLG0 + CLLJ0
   70 RETURN
      END
      SUBROUTINE TREFTZ(Y,DELTA,CMU,GAMB,ALFINF,NROWS,LIKE)
      DIMENSION Y(40),DELTA(40),CMU(40),GAMB(40),ALFINF(40)
      DIMENSION E(40),B(40),C(40),DP(40),DM(40),DGAM(40)
```

```
      LOGICAL Z1,Z2,Z3,Z4,Z5,Z6,ZT1,ZT2
C
C  FIND COEFFICIENTS OF CURVES
      J=0
      DO 60 I=1,NROWS
      Z1=I.EQ.1
      Z2=LIKE.GT.0.AND.I.EQ.NROWS
      Z3=(LIKE.LE.0.AND.I.EQ.NROWS).OR.(LIKE.GT.0.AND.I.NE.NROWS.AND.
     1Y(I).GT.0.0.AND.Y(I+1).LT.0.0)
      Z4=LIKE.GT.0.AND.I.GT.1.AND.Y(I).LT.0.0.AND.Y(I-1).GT.0.0
      Z5=(I.NE.NROWS).AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.
     1(CMU(I).GT.0.0.AND.CMU(I+1).LT.0.0001))
      Z6=(I.GT.1).AND.((CMU(I).GT.0.0.AND.CMU(I-1).LT.0.0001).OR.
     1(CMU(I).LT.0.0001.AND.CMU(I-1).GT.0.0))
      D1=SQRT(1.0-ABS(Y(I)))
      ZT1=Z1.OR.Z4.OR.Z6
      ZT2=Z2.OR.Z3.OR.Z5
      IF(J.EQ.1)GO TO 10
      IF(ZT1)J=1
      IF(ZT1)GO TO 40
      J=0
   10 CONTINUE
      IF(ZT2) GO TO 30
      D2=SQRT(1.0-ABS(Y(I+1)))
      D3=SQRT(1.0-ABS(Y(I-1)))
      G1=GAMB(I)
      G2=GAMB(I+1)
      G3=GAMB(I-1)
      D13=D1**3
      D23=D2**3
      D33=D3**3
      D15=D1**5
      D25=D2**5
      D35=D3**5
      F1=D23*D35-D33*D25
      F2=D13*D35-D33*D15
      F3=D13*D25-D23*D15
      E(I)=(G1*F1-G2*F2+G3*F3)/(D1*F1-D2*F2+D3*F3)
      B(I)=((G1-E(I)*D1)*D25-(G2-E(I)*D2)*D15)/(D13*D25-D23*D15)
      C(I)=(G1-E(I)*D1-B(I)*D13)/D15
      IF(J.EQ.0) GO TO 40
   20 E(I-1)=E(I)
      B(I-1)=B(I)
      C(I-1)=C(I)
      J=0
      GO TO 40
   30 E(I)=E(I-1)
      B(I)=B(I-1)
      C(I)=C(I-1)
   40 CONTINUE
      IF(Y(I).LT.0.0) GO TO 50
      DP(I)=SQRT(ABS(D1**2-DELTA(I)))
      DM(I)=SQRT(D1**2+DELTA(I))
      GO TO 60
   50 DP(I)=SQRT(D1**2+DELTA(I))
      DM(I)=SQRT(D1**2-DELTA(I))
   60 CONTINUE
C
C  CALCULATE STRENGTH OF DISCRETE VORTICIES
      DO 80 I=1,NROWS
      IF(I.LT.NROWS.AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.
     1(CMU(I).GT.0.0.AND.CMU(I+1).LT.0.0001))) GO TO 70
      DGAM(I)=0.0
      GO TO 80
   70 DGAM(I)=(E(I+1)-E(I))*DM(I)+(B(I+1)-B(I))*DM(I)**3+(C(I+1)-C(I))
     1*DM(I)**5
   80 CONTINUE
C
C  CALCULATE DOWNWASH
      DO 100 I=1,NROWS
      ALFINF(I)=0.0
      F2=SQRT(1.0-Y(I))
      F4=SQRT(1.0+Y(I))
      DO 90 J=1,NROWS
      IF(Y(J).LT.0.0) F2=F4
      CON=1.0
      IF(Y(J).LT.0.0) CON=-1.0
      F1=ALOG(ABS((DP(J)-F2)*(DM(J)+F2))/((DP(J)+F2)*(DM(J)-F2))))
      S1=CON*F1/F2
      S2=(2.0*(DP(J)-DM(J))+F2*F1)*CON
      S3=CON*(2.0*(DP(J)**3-DM(J)**3)/3.0)+F2**2*S2
      DELG=-DGAM(J)/(Y(I)-Y(J)+DELTA(J))
      IF(LIKE.GT.0) GO TO 90
      F3=ALOG(ABS(((DP(J)-F4)*(DM(J)+F4))/((DP(J)+F4)*(DM(J)-F4))))
      S2P=2.0*(DP(J)-DM(J))+F4*F3
      IF(LIKE.LT.0) SY=-1.0
      IF(LIKE.EQ.0) SY=1.0
      S1=S1+SY*F3/F4
      S2=S2+SY*S2P
      S3=S3+SY*((2.0*(DP(J)**3-DM(J)**3)/3.0)+F4**2*S2P)
      DELG=DELG-SY*DGAM(J)/(-Y(I)-Y(J)+DELTA(J))
   90 ALFINF(I)=ALFINF(I)+0.1591549*(0.5*E(J)*S1+1.5*B(J)*S2+2.5*C(J)*
     1S3+DELG)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE TABLE(CLO,CLA,CMO,CMA,CLLO,CLLA,CDIO,CDIX,CDIA2,CJ,
     1                 CNIO,CNIX,CNIA2,CNJ,CYO,CYX,CYA2,MCASE)
C
C  THIS SUBROUTINE COMPUTES AND PRINTS A TABLE OF TOTAL COEFFICIENTS
C  FOR A RANGE OF ANGLES OF ATTACK
```

```
C
C     COMPUTE THE COEFFICIENTS
         WRITE(6, 10 ) MCASE
   10 FORMAT(1H1,31X,14(4H****)     /32X,27H*  TABULATED TOTAL COEFFICI,
     1        23HENTS FOR COMPOSITE CASE, I3, 3H  */ 32X,14(4H****)//
     2   4X,5HALPHA,10X,3HCCL,8X,6HCCL**2,5X,7HCCM(MC),4X,3HCLL,13X,
     3   5HCDIT2,6X,3HCCT,13X,3HCNI,8X,2HCN,9X,3HCCY)
         ALPHA = -11.00
         DO 60  M = 1,41
   20 ALPHA = ALPHA + 1.00
         ALPHA2 = ALPHA * ALPHA
C     LINEAR COEFFICIENTS
   30 CL = CL0 + CLA * ALPHA
         CL2 = CL * CL
         CM = CM0 + CMA * ALPHA
         CLL = CLL0 + CLLA * ALPHA
C     NONLINEAR COEFFICIENTS
   40 CDI = CDI0 + CDIX * ALPHA + CDIA2 * ALPHA2
         CT = CJ - CDI
         CNI = CNI0 + CNIX * ALPHA + CNIA2 * ALPHA2
         CN = CNJ + CNI
         CY = CY0 + CYX * ALPHA + CYA2 * ALPHA2
C
C     PRINT THE TABLE
         WRITE(6, 50 ) ALPHA,CL,CL2,CM,CLL,CDI,CT,CNI,CN,CY
   50 FORMAT(1H , F10.6,5H  *  ,4F11.7,5H  *  ,2F11.7,5H  *  ,3F11.7)
   60 CONTINUE
         RETURN
         END
         SUBROUTINE FUNDER(EPS,CP0,CPA,CPR0,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,
     1   AREA,CL9,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,
     2       NROWS,ISYMM,XL,TL,XMC)
C
C     THIS SUBROUTINE CONTROLS CALCULATION OF ALL AERODYNAMIC COEFFICIENTS
C     AND STABILITY DERIVATIVES FOR THE FUNDAMENTAL CASES
C
         COMMON/SOLV1/CP(600,10)
         DIMENSION CP0(NEWMAX),CPA(NEWMAX),CPR0(600),CPRA(600),CPP(600)
         DIMENSION DEL(600),EPS(600,10),EP(600)
         DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
         DIMENSION NW(40),IJ(40)
C
C     FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN HAS STORED
         IREAD = NMAX + NJT
         IFUDGE = 0
         CLLP = 0.00
         CNP2 = 0.00
         CNP0 = 0.00
C     CALCULATE THE CHORDWISE LOADING
   10 CALL STG3FC(NEWMAX)
C
C     DEFINE THE DUMMY CP ARRAYS FOR THE ALPHA CASE
C     AND THE CP ARRAY FOR THE ROLLING FUNDAMENTAL CASE
         DO 20 I = 1,NEWMAX
         CPA(I) = 0.0
         CPRA(I) = 0.0
         CPP(I) = CP(I,NCASES)
   20 CONTINUE
         DO 120 N = 1,NCASES
         LCASE = N
         IREAD = IREAD + 1
         FIND(1'IREAD)
C
C     CALCULATE THE SPANWISE AND TOTAL LOADING
   30 IF(N .EQ. NCASES)
     1        CALL STG3FS(DUM1,DUM2,DUM3,CLLP,NEWMAX,NOALFA,LCASE)
C
C     READ THE FIRST RUN SOLUTION
         READ(1'IREAD) CP0
C
C     DEFINE CP AND EP ARRAYS FOR THE PRESENT SECOND RUN FUNDAMENTAL CASE
         DO 40  I = 1,NEWMAX
         CPR0(I) = CP(I,LCASE)
         EP(I) = EPS(I,LCASE)
   40 CONTINUE
C
C     CALCULATE THE STABILITY DERIVATIVES
         CALL SUMIT1(CP0,CPA,CPP,CPR0,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,
     1     XMC,AREA,DUM1,DUM2,CNP2,CLLR0,DUM3,DUM5,DUM6,CYP2,
     2     NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
         IF(N .LT. NCASES) GO TO 50
         CNP0 = DUM1
         GO TO 100
   50 CALL SUMIT2(CP0,CPA,CPR0,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,
     1        XMC,CNR0,DUM1,DUM2,CNR20,DUM3,DUM4,CYR0,DUM5,DUM6,
     2        CYR20,DUM7,DUM8,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C     PRINT THE STABILITY DERIVATIVE DATA
   60 WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,32X,13(4H****), 3H*** / 33X,
     1        49H*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE, I3,
     2        3H  * / 33X,13(4H****), 3H*** /)
         WRITE(6, 80 ) CLLR0
   80 FORMAT(1H0/ 25X, 44HROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO,
     1        15H YAWING, CLLR =, F12.7)
         WRITE(6, 90 ) CNR0,CNR20
   90 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     1       53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS //
     2        49X, 25HCN(R) = CNR*R + CNR2*R**2 //
     3        49X, 13HWHERE    CNR =,F14.9 / 56X, 6HCNR2 =,F12.7)
         WRITE(6,95) CYR0,CYR20
```

```
   95 FORMAT(1H0/ 23X,43HSIDE FORCE COEFFICIENT DUE TO YAWING, CY(R),
     1        29H MAY BE CALCULATED AS FOLLOWS//
     2        49X,25HCY(R) = CYR*R + CYR2*R**2//
     3        49X,13HWHERE   CYR =,F14.9/ 56X,6HCYR2 =,F12.7)
      GO TO 120
  100 WRITE(6, 70 ) LCASE
      WRITE(6, 110 ) CLLP,CNP2,CYP2
  110 FORMAT(1H0/ 28X, 38HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     1        16H ROLLING, CLLP =, F12.7 ///
     2        17X, 42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     3        44H ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS //
     4        53X, 17HCN(P) = CNP2*P**2 //
     5        53X, 13HWHERE  CNP2 =,F12.7///
     6        30X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),
     7        29H MAY BE CALCULATED AS FOLLOWS//
     8        53X,17HCY(P) = CYP2*P**2//
     9        53X,13HWHERE  CYP2 =,F12.7)
  120 CONTINUE
      RETURN
      END
      SUBROUTINE COMDER(EPS,CP0,CPA,CPR0,CPRA,CPP,CPREAD,DEL,CHORD,Y,
     1    CMU,DELTA,AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,
     2    NCASES,NROWS,ISYMM,XL,TL,XMC)
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF STABILITY DERIVATIVES
C  FOR ALL COMPOSITE CASES
C
      COMMON/SOLV1/CP(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      DIMENSION CP0(NEWMAX),CPA(NEWMAX),CPR0(600),CPRA(600),CPP(600),
     1          CPREAD(NEWMAX)
      DIMENSION DEL(600),EPS(600,10),EP(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  CYCLE THROUGH ALL COMPOSITE CASES
      IFUDGE = 1
      DO 100 M = 1,NCC
      MCASE = M
      NC1 = NCASES - 1
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED
      IREAD = NMAX + NJT + 1
      FIND(1'IREAD)
C
C  DEFINE CP AND EP ARRAYS FOR THIS SECOND RUN COMPOSITE CASE
C  NOTE THAT CPP HAS PREVIOUSLY BEEN DEFINED IN FUNDER
      DO 20 I = 1,NEWMAX
      CPR0(I) = 0.00
      EP(I) = 0.00
      CPRA(I) = CP(I,1)
      DO 10 N = 1,NC1
      CPR0(I) = CPR0(I) + FACTOR(N,M) * CP(I,N)
      EP(I) = EP(I) + FACTOR(N,M) * EPS(I,N)
   10 CONTINUE
   20 CONTINUE
C
C  DEFINE CP FOR THE FIRST RUN COMPOSITE CASE
      IF(M .EQ. 1) READ(1'IREAD) CPA
      DO 30 I = 1,NEWMAX
      CP0(I) = 0.00
   30 CONTINUE
      IREAD = IREAD - 1
      DO 60 N = 1,NC1
      IREAD = IREAD + 1
      FIND(1'IREAD)
      IF(FACTOR(N,M) .EQ. 0.0) GO TO 60
   40 READ(1'IREAD) CPREAD
      DO 50 I = 1,NEWMAX
      CP0(I) = CP0(I) + FACTOR(N,M) * CPREAD(I)
   50 CONTINUE
   60 CONTINUE
C  CALCULATE THE COMPOSITE CASE DERIVATIVES
   70 CALL SUMIT1(CP0,CPA,CPP,CPR0,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,
     1        XMC,AREA,CNP0,CNPA,CNP2,CLLR0,CLLRA,CYP0,CYPA,CYP2,
     2        NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
   80 CALL SUMIT2(CP0,CPA,CPR0,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,
     1 XMC,CNR0,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,CYR0,CYRA,CYRA2,
     1 CYR20,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C  PRINT A SUMMARY TABLE OF ALL STABILITY DERIVATIVES
   90 CALL STABLE(CLQ,CMQ,CMQMC,CLLP,CNP0,CNPA,CNP2,CYP0,CYPA,CYP2,
     1        CLLR0,CLLRA,CNR0,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,
     1        CYR0,CYRA,CYRA2,CYR20,CYR2A,CYR2A2,MCASE)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE SUMIT1(CP0,CPA,CPP,CPR0,CPRA,DEL,EPS,CHORD,Y,DELTA,CMU,
     1  XL,TL,XMC,AREA,CNP0,CNPA,CNP2,CLLR0,CLLRA,CYP0,CYPA,CYP2,
     2  NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)
C
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE
C  THE TERMS OF CNP AND CLLR DERIVATIVES
C
      COMMON /DERIV/ U0(40),CLQ,CMQ,CMQMC
      DIMENSION CP0(NEWMAX),CPA(NEWMAX),CPP(600),CPR0(600),CPRA(600)
      DIMENSION DEL(600),EPS(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  INITIALIZE THE DERIVATIVE TERMS
   10 CNP0 = 0.00
```

```
      CNPA = 0.00
      CNP2 = 0.00
      CYP0 = 0.00
      CYPA = 0.00
      CYP2 = 0.00
      CLLR0 = 0.00
      CLLRA = 0.00
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      I = 0
      DO 100 K = 1,NROWS
C
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE
C  LEADING EDGE CONTRIBUTION
      I = I + 1
C  YAWING DUE TO ROLLING
   20 TERMP = DEL(I) * (CPP(I) + 0.50*CPP(I+1))
      EP = -EPS(I) / (57.295779 * U0(K))
      DP0 = TERMP * EP
      DPA = TERMP
      SP0 = 0.3490658 * DEL(I) * CPP(I) * CP0(I)
      SPA = 0.3490658 * DEL(I) * CPP(I) * CPA(I)
      SP2 = 0.1745329 * DEL(I) * CPP(I)**2
C  ROLLING DUE TO YAWING
   30 CLG0 = DEL(I) * (CPR0(I) + 0.50*CPR0(I+1))
      CLGA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))
C
C  REGULAR EVD CONTRIBUTIONS
   40 NWK = NW(K)
      DO 70  L = 2,NWK
      I = I + 1
      CPP1 = CPP(I+1)
      CPR01 = CPR0(I+1)
      CPRA1 = CPRA(I+1)
      IF(L .LT. NWK) GO TO 50
      CPP1 = 0.00
      CPR01 = 0.00
      CPRA1 = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      IJK = IJ(K)
      CPP1 = CPP(IJK)
      CPR01 = CPR0(IJK)
      CPRA1 = CPRA(IJK)
   50 EP = -EPS(I) / (57.295779 * U0(K))
      TERMP = 0.50 * DEL(I) * (CPP(I) + CPP1)
      DP0 = DP0 + TERMP * EP
      DPA = DPA + TERMP
   60 CLG0 = CLG0 + 0.50 * DEL(I) * (CPR0(I) + CPR01)
      CLGA = CLGA + 0.50 * DEL(I) * (CPRA(I) + CPRA1)
   70 CONTINUE
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      FACTOR = CHORD(K) * Y(K) * DELTA(K)
      FACTO = CHORD(K) * DELTA(K) * TL(K)
      FACT = FACTO * (XL(K)-XMC)
   80 CNP0 = CNP0 + (DP0 - SP0) * FACTOR - SP0 * FACT
      CNPA = CNPA + (DPA/57.295779 - SPA) * FACTOR - SPA * FACT
      CNP2 = CNP2 + SP2 * FACTOR + SP2 * FACT
      CYP0 = CYP0 + SP0 * FACTO
      CYPA = CYPA + SPA * FACTO
      CYP2 = CYP2 + SP2 * FACTO
   90 CLLR0 = CLLR0 + CLG0 * FACTOR
      CLLRA = CLLRA + CLGA * FACTOR
  100 CONTINUE
C
C  PUT THE TERMS IN FINAL FORM
      FACTOR = 1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA
  110 CNP0 = FACTOR * CNP0
      CNPA = FACTOR * CNPA
      CNP2 = -FACTOR * CNP2
      CLLR0 = -FACTOR * CLLR0
      CLLRA = -FACTOR * CLLRA
      FACTOR = 2.00 * FACTOR
      CYP0 = FACTOR * CYP0
      CYPA = FACTOR * CYPA
      CYP2 = FACTOR * CYP2
      IF(IFUDGE .NE. 0) GO TO 115
      CNP0 = 0.00
      CYP0 = 0.00
  115 IF(ISYMM .GE. 0) GO TO 120
      CNP2 = 0.00
      CYP2 = 0.00
  120 RETURN
      END
      SUBROUTINE SUMIT2(CP0,CPA,CPR0,CPRA,DEL,EPS,CHORD,Y,DELTA,AREA,
     1  CMU,XL,TL,XMC,CNR0,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,CYR0,CYRA,
     2  CYRA2,CYR20,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)
C
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE
C  THE TERMS OF THE YAWING AND SIDE FORCE COEFFICIENTS DUE TO YAWING
C
      COMMON /DERIV/ U0(40),CLQ,CMQ,CMQMC
      DIMENSION CP0(NEWMAX),CPA(NEWMAX),CPR0(600),CPRA(600)
      DIMENSION DEL(600),EPS(600)
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)
      DIMENSION NW(40),IJ(40)
C
C  INITIALIZE THE YAWING COEFFICIENT TERMS
   10 CNR0 = 0.00
      CNRA = 0.00
```

```
      CNRA2 = 0.00
      CNR20 = 0.00
      CNR2A = 0.00
      CNR2A2 = 0.00
      CYRO = 0.00
      CYRA = 0.00
      CYRA2 = 0.00
      CYR20 = 0.00
      CYR2A = 0.00
      CYR2A2 = 0.00
C
      I = 0
      DO 100 K = 1,NROWS
C
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE
C  LEADING EDGE CONTRIBUTIONS
      I = I + 1
   20 TERMO = DEL(I) * (CPRO(I) + 0.50*CPRO(I+1))
      TERMA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))
      EP = -EPS(I) / (57.295779 * UO(K))
      DRO = TERMO * EP
      DRA = TERMO/57.295779 + TERMA * EP
      DRA2 = TERMA
   30 SRO = 0.3490658 * DEL(I) * CPRO(I) * CPO(I)
      SRA = 0.3490658 * DEL(I) * (CPRO(I)*CPA(I) + CPRA(I)*CPO(I))
      SRA2 = 0.3490658 * DEL(I) * (CPRA(I)*CPA(I))
      IF(ISYMM .LT. 0) GO TO 40
      SR20 = 0.1745329 * DEL(I) * CPRO(I)**2
      SR2A = 0.3490658 * DEL(I) * CPRO(I)*CPRA(I)
      SR2A2 = 0.1745329 * DEL(I) * CPRA(I)**2
C
C  REGULAR EVD CONTRIBUTIONS
   40 NWK = NW(K)
      DO 70  L = 2,NWK
      I = I + 1
      CPRO1 = CPRO(I+1)
      CPRA1 = CPRA(I+1)
      IF(L .LT. NWK) GO TO 50
      CPRO1 = 0.00
      CPRA1 = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 50
      IJK = IJ(K)
      CPRO1 = CPRO(IJK)
      CPRA1 = CPRA(IJK)
   50 EP = -EPS(I) / (57.295779 * UO(K))
C  REGULAR EVD CONTRIBUTIONS
      TERMO = 0.50 * DEL(I) * (CPRO(I)+CPRO1)
   60 TERMA = 0.50 * DEL(I) * (CPRA(I)+CPRA1)
      DRO = DRO + TERMO * EP
      DRA = DRA + TERMO/57.295779 + TERMA * EP
      DRA2 = DRA2 + TERMA
   70 CONTINUE
C
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE
      FACTOR = CHORD(K) * Y(K) * DELTA(K)
      FACTO = CHORD(K) * TL(K) * DELTA(K)
      FACT = FACTO * (XL(K)-XMC)
   80 CNRO = CNRO + (DRO-SRO) * FACTOR - SRO * FACT
      CNRA = CNRA + (DRA-SRA) * FACTOR - SRA * FACT
      CNRA2 = CNRA2 + (DRA2/57.295779-SRA2) * FACTOR - SRA2 * FACT
      CYRO = CYRO + SRO * FACTO
      CYRA = CYRA + SRA * FACTO
      CYRA2 = CYRA2 + SRA2 * FACTO
      IF(ISYMM .LT. 0) GO TO 100
   90 CNR20 = CNR20 - SR20 * FACTOR -SR20 * FACT
      CNR2A = CNR2A - SR2A * FACTOR - SR2A * FACT
      CNR2A2 = CNR2A2 - SR2A2 * FACTOR - SR2A2 * FACT
      CYR20 = CYR20 + SR20 * FACTO
      CYR2A = CYR2A + SR2A * FACTO
      CYR2A2 = CYR2A2 + SR2A2 * FACTO
  100 CONTINUE
C
C  PUT THE TERMS IN FINAL FORM
      FACTOR = 1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA
  110 CNRO = FACTOR * CNRO
      CNRA = FACTOR * CNRA
      CNRA2 = FACTOR * CNRA2
      CNR20 = FACTOR * CNR20
      CNR2A = FACTOR * CNR2A
      CNR2A2 = FACTOR * CNR2A2
      FACTOR = 2.00 * FACTOR
      CYRO = FACTOR * CYRO
      CYRA = FACTOR * CYRA
      CYRA2 = FACTOR * CYRA2
      CYR20 = FACTOR * CYR20
      CYR2A = FACTOR * CYR2A
      CYR2A2 = FACTOR * CYR2A2
  120 RETURN
      END
      SUBROUTINE STABLE(CLQ,CMQ,CMQMC,CLLP,CNPO,CNPA,CNP2,CYPO,CYPA,
     1       CYP2,CLLRO,CLLRA,CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,
     2       CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2,MCASE)
C
C  THIS SUBROUTINE CALCULATES AND PRINTS A COMPLETE SUMMARY TABLE
C  OF ALL STABILITY DERIVATIVE DATA FOR EACH COMPOSITE CASE
C
      COMMON/COMPOS/FACTOR(10,24),NCC
C
C  PRINT ALL CONSTANT DERIVATIVES
      WRITE(6, 10 ) MCASE
```

```
   10 FORMAT(1H1,33X,13(4H****),1H* /  34X,
     1        47H*  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE, I3,
     2        3H  */ 34X,13(4H****),1H* /)
      WRITE(6, 20 ) (N,N=1,10), (FACTOR(N,MCASE),N=1,10)
   20 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/ 10X,9(4X,2HA(,I1,1H),
     1      2X),3X,2HA(,I2,1H) / 10X,10F10.6)
      WRITE(6, 30 ) CLQ,CMQ,CMQMC
   30 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,
     1        17H ABOUT XCG, CLQ =,F10.6 /
     2        14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,
     3        33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /
     4        16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,
     5        35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)
      WRITE(6, 40 ) CLLP,CNP0,CNPA,CNP2
   40 FORMAT(1H0/28X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     1        16H ROLLING, CLLP =,F12.7 ///
     2        17X,51HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING,,
     3        35H CN(P) MAY BE CALCULATED AS FOLLOWS /
     4        48X,25HCN(P) = CNP*P + CNP2*P**2 /
     5        42X,31HWHERE   CNP = CNP0 + CNPA*ALPHA /
     6        52X, 6HCNP0 =,F12.7/52X, 6HCNPA =,F12.7/52X,6HCNP2 =,F12.7)
      WRITE(6,45) CYP0,CYPA,CYP2
   45 FORMAT(1H0/26X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),
     1        29H MAY BE CALCULATED AS FOLLOWS/
     2        48X,25HCY(P) = CYP*P + CYP2*P**2/
     3        42X,31HWHERE   CYP = CYP0 + CYPA*ALPHA/
     4        52X, 6HCYP0 =,F12.7/52X,6HCYPA =,F12.7/52X,6HCYP2 =,F12.7)
      WRITE(6, 50 ) CLLR0,CLLRA
   50 FORMAT(1H0/15X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,
     1        52H YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS /
     2        48X,26HCLLR = CLLR0 + CLLRA*ALPHA /
     3        46X,15HWHERE   CLLR0 =,F13.7 / 56X,7HCLLRA =,F13.7)
      WRITE(6, 60 ) CNR0,CNRA,CNRA2,CNR20,CNR2A,CNR2A2
   60 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,
     1        53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS /
     2        48X,25HCN(R) = CNR*R + CNR2*R**2 /
     3        36X,48HWHERE   CNR = CNR0 + CNRA*ALPHA + CNRA2*ALPHA**2 /
     4        52X,6HCNR0 =,F13.7/52X,6HCNRA =,F13.7/51X,7HCNRA2 =,F13.7/
     5     36X,51HAND    CNR2 = CNR20 + CNR2A*ALPHA + CNR2A2*ALPHA**2/
     6   51X,7HCNR20 =,F13.7/51X,7HCNR2A =,F13.7/50X,8HCNR2A2 =,F13.7)
      WRITE(6,65) CYR0,CYRA,CYRA2,CYR20,CYR2A,CYR2A2
   65 FORMAT(1H0/13X,39HSIDE FORCE COEFFICIENT ABOUT XMC DUE TO,
     1        53H YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS /
     2        48X,25HCY(R) = CYR*R + CYR2*R**2 /
     3        36X,48HWHERE   CYR = CYR0 + CYRA*ALPHA + CYRA2*ALPHA**2 /
     4        52X,6HCYR0 =,F13.7/52X,6HCYRA =,F13.7/51X,7HCYRA2 =,F13.7/
     5     36X,51HAND    CYR2 = CYR20 + CYR2A*ALPHA + CYR2A2*ALPHA**2/
     6   51X,7HCYR20 =,F13.7/51X,7HCYR2A =,F13.7/50X,8HCYR2A2 =,F13.7)
C
C  PRINT TABLE OF DERIVATIVE TERMS WHICH DEPEND ON ALPHA
      WRITE(6, 70 )
   70 FORMAT(1H1,32X,3H***,10(5H*****),/,
     1 33X,53H* VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK *,/,
     2 33X,3H***,10(5H*****),//,
     3 9X,5HALPHA,12X,3HCNP,10X,4HCNP2,14X,3HCYP,10X,4HCYP2,
     4 14X,4HCLLR, 9X,3HCNR,10X,4HCNR2)
      ALPHA = -11.00
      DO 120 M = 1,41
   80 ALPHA = ALPHA + 1.00
      CNP = CNP0 + CNPA * ALPHA
        CYP = CYP0 + CYPA * ALPHA
   90 CLLR = CLLR0 + CLLRA * ALPHA
      CNR = CNR0 + CNRA * ALPHA + CNRA2 * ALPHA**2
  100 CNR2 = CNR20 + CNR2A * ALPHA + CNR2A2 * ALPHA**2
      WRITE(6, 110 ) ALPHA,CNP,CNP2,CYP,CYP2,CLLR,CNR,CNR2
  110 FORMAT(1H , 5X,F10.6,5H  *  ,2F13.7,5H  *  ,2F13.7,5H  *  ,
     1          3F13.7)
  120 CONTINUE
      RETURN
      END
      SUBROUTINE STAGE4
C
C  THIS PROGRAM CONTROLS THE EXECUTION OF UTILITY ROUTINES AND
C  BOUNDARY CONDITION SETUP FOR STABILITY DERIVATIVE RUNS
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/SOLV1/CP(600,10)
      DIMENSION CPREAD(600)
C
C  ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING
      IF(LOGIC .GT. 1) GO TO 60
   10 IF(NCASES .LT. 10) GO TO 30
      WRITE(6, 20 )
   20 FORMAT(1H0//14X,44HFUNDAMENTAL CASE 10 HAS BEEN REPLACED BY THE,
     1       47H FOLLOWING CASE FOR DERIVATIVES DUE TO PITCHING)
      GO TO 40
   30 NCASES = NCASES + 1
   40 CALL BCPICH(XCG,CREF,XI,DEL,EPS,BETA,CHORD,KK,THETA,THS,TST,HL,
     1      NW,IW,NJ,IJ,NWT,NMAX,NROWS,NCASES)
   50 CALL OUT2(NCASES)
      GO TO 100
C
C  SAVE THE FIRST RUN SOLUTION ON UNIT 1
```

```
   60 ISIZE = NEWMAX
      CALL SAVECP(CP,CPREAD,NMAX,NJT,ISIZE,NCASES)
C
C  DEFINE THE FUNDAMENTAL CASES FOR YAWING DERIVATIVES
      NC1 = NCASES - 1
   70 CALL BCYAW(EPS,BETA,THETA,THS,Y,KK,NWT,NMAX,NROWS,NC1)
C  DEFINE THE LAST FUNDAMENTAL CASE FOR ROLLING DERIVATIVES
   80 CALL BCROLL(EPS,BETA,THETA,THS,TST,Y,NW,NWT,NMAX,NROWS,NCASES)
C
C  PRINT THE FUNDAMENTAL CASE GEOMETRY
      IF(IPRINT .GE. 0) GO TO 100
      DO 90  N = 1,NCASES
      LCASE = N
      CALL OUT2(LCASE)
   90 CONTINUE
  100 RETURN
      END
      SUBROUTINE OUT2(LCASE)
C
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE
C  SECTIONAL METHOD INPUT
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(20)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
C
C  PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS
      IF(LCASE .GT. 1) GO TO 60
   10 WRITE(6, 20 )TITLE
   20 FORMAT(1H1,39X,10(4H****)/
     1    40X,40H*  EVD  JET - WING  COMPUTER  PROGRAM  */
     2    40X,10(4H****)///20X,20A4)
      CMA = CMAC * SPA / 2.0
   30 WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,
     1             ARATIO,XCG,XC
   40 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /
     1         41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /
     2         41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /
     3         41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /
     4         42X,5HXCG =,2F15.6)
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,
     1             IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX
   50 FORMAT(1H0/ 48X,7HNROWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /
     1              48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /
     2              47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /
     3              47X,8HIHINGE =,I3,7X,I3 ///
     4              43X,25HNUMBER OF WING ELEMENTS =,I4 /
     5              43X,25HNUMBER OF JET ELEMENTS  =,I4 /
     6              42X,26HTOTAL NUMBER OF ELEMENTS =,I4)
   60 J = 0
      JJ = NWT
C
C  PRINT FUNDAMENTAL CASE HEADER
      WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,23X,1H*,19(4H****)/
     1    24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,
     2    17H FUNDAMENTAL CASE,I3,3H  */24X,1H*,19(4H****))
      ILINES = 3
      DO 260 K = 1,NROWS
C
C  PRINT SECTIONAL DATA
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)
   80 FORMAT(1H0,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,
     2 2X,7HTANLE =,F10.6)
C
C  PRINT CHORDWISE DATA ON WING
      NWK = NW(K)
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)
   90 FORMAT(1H0,21HWING ELEMENTS    NW =,I3,5X, 7HTWIST =,F10.6,5X,
     1             4HHL =,F10.6,5X, 9HTHETA S =,F10.6)
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)
  100 FORMAT(1H ,14X,2HXB,  10F11.6/17X,10F11.6)
      IF(LCASE .GT. 1) GO TO 130
      WRITE(6, 110 ) (XI(J+L),L=1,NWK)
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)
  130 IF(ICT(K) .EQ. 0) GO TO 150
      ICK = ICT(K)
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)
  140 FORMAT(1H ,10X,6HCAMBER,10F11.6/17X,10F11.6)
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))
      J = J + NWK
      IL = 1
      IF(NWK .GT. 9) IL = 2
      ILINES = ILINES + 4 + 4*IL
```

```
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
C
C  PRINT CHORDWISE DATA ON JET
      NJK = NJ(K)
      IF(NJK .GT. 0) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)
      ILINES = ILINES + 1
      GO TO 230
  200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)
  210 FORMAT(1H0,1X,20HJET ELEMENTS     NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,
     1     F10.6,5X,6HACTE =,F10.6,5X,7HTHETA =,F10.6)
      WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)
      IF(LCASE .GT. 1) GO TO 220
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)
      JJ = JJ + NJK
      IL = 1
      IF(NJK .EQ. 10) IL = 2
      ILINES = ILINES + 1 + 3 * IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
  230 IF(K .EQ. NROWS) GO TO 260
      NWK1 = NW(K+1)
      IL = 1
      IF(NWK1 .GT. 9) IL = 2
      NEXT = 4 + 4*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
      NJK1 = NJ(K+1)
      IL = 1
      IF(NJK1 .EQ. 10) IL = 2
      NEXT = NEXT + 1
      IF(NJK1 .EQ. 0) GO TO 240
      NEXT = NEXT + 1 + 3*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
  240 IF((55-ILINES) .GE  NEXT) GO TO 260
      WRITE(6, 250 )
  250 FORMAT(1H1)
      ILINES = 1
  260 CONTINUE
      RETURN
      END
      SUBROUTINE SAVECP(CP,DUMMY,NMAX,NJT,ISIZE,NCASES)
C
C  THIS SUBROUTINE SAVES THE CP SOLUTION FOR ALL FUNDAMENTAL CASES
C  OF THE FIRST STABILITY DERIVATIVE RUN BY STORING ON DIRECT ACCESS
C
      DIMENSION CP(600,10),DUMMY(ISIZE)
C
C  FIND THE PROPER PLACE TO WRITE THE OLD SOLUTION
   10 IWRITE = NMAX + NJT
C       FIND(1'IWRITE+1)                       *** COMMENTED OUT BY JAC***
C  DEFINE THE DUMMY ARRAY
      DO 30 N = 1,NCASES
      IWRITE = IWRITE + 1
      DO 20 I = 1,ISIZE
      DUMMY(I) = CP(I,N)
   20 CONTINUE
C
C  SAVE THE DATA
      WRITE(1'IWRITE) DUMMY
   30 CONTINUE
      RETURN
      END
      SUBROUTINE BCPICH(XCG,CREF,XI,DEL,EPS,BETA,C,KK,THETA,THS,TST,HL,
     1                  NW,IW,NJ,IJ,NWT,NMAX,NROWS,N)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR
C  THE PITCHING RATE DERIVATIVE FUNDAMENTAL CASE
C
      DIMENSION XI(600),DEL(600),C(40),KK(600)
      DIMENSION NW(40),IW(40),NJ(40),IJ(40)
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION TST(40,10),HL(40,10)
C
C  DEFINE THE CAMBER ANGLES WHICH RESULT FROM PITCHING
      DO 20 I = 1,NWT
      KKI = KK(I)
   10 EPS(I,N) = 2.00 * (XI(I)+(DEL(I)/2.0)*C(KKI)-XCG) / CREF
      BETA(I,N) = 0.00
   20 CONTINUE
      IF(NWT .EQ. NMAX) GO TO 50
      NWT1 = NWT + 1
      DO 40 I = NWT1,NMAX
   30 EPS(I,N) = 0.00
      BETA(I,N) = 0.00
   40 CONTINUE
C
C  DEFINE THE JET ANGLES WHICH RESULT FROM PITCHING
   50 DO 70 K = 1,NROWS
   60 THETA(K,N) = 0.00
      IJK = IJ(K)
      IF(NJ(K) .GT. 0) THETA(K,N) = 2.00 * (XI(IJK) - XCG) / CREF
      THS(K,N) = 0.00
      TST(K,N) = 0.00
      HL(K,N) = 0.00
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BCROLL(EPS,BETA,THETA,THS,TST,Y,
     1              NW,NWT,NMAX,NROWS,NCASES)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR
C  THE ROLLING RATE DERIVATIVE FUNDAMENTAL CASE
C
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10),
     1          TST(40,10)
      DIMENSION Y(40),NW(40)
C
C  DEFINE THE TWIST AND CAMBER ANFLES WHICH RESULT FROM ROLLING
      N = NCASES
      I = 0
      DO 40 K = 1,NROWS
   10 TST(K,N) = Y(K)
      THETA(K,N) = TST(K,N)
      THS(K,N) = 0.00
      NWK = NW(K)
      DO 30 L = 1,NWK
      I = I + 1
   20 EPS(I,N) = TST(K,N)
   30 CONTINUE
   40 CONTINUE
C
C  DEFINE THE ANGLES ON THE JET
      IF(NMAX .EQ. NWT) RETURN
      NWT1 = NWT + 1
      DO 60 I = NWT1,NMAX
   50 EPS(I,N) = 0.00
   60 CONTINUE
      RETURN
      END
      SUBROUTINE BCYAW (EPS,BETA,THETA,THS,Y,KK,
     1              NWT,NMAX,NROWS,NCASES)
C
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR ALL OF THE
C  YAWING RATE DERIVATIVE FUNDAMENTAL CASES
C
      COMMON /DERIV/ U0(40),CLQ,CMQ,CMQMC
      DIMENSION EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      DIMENSION Y(40),KK(600)
C
C  DEFINE THE SECTIONAL NORMALIZED VELOCITY INDUCED BY YAWING
      DO 10 K = 1,NROWS
      U0(K) = Y(K) / 57.295779
   10 CONTINUE
C
C  DEFINE THE ANGLES FOR ALL FUNDAMENTAL CASES
      DO 80  N = 1,NCASES
      DO 30 I = 1,NWT
      KKI = KK(I)
   20 EPS(I,N) = -U0(KKI) * EPS(I,N)
      BETA(I,N) = 0.00
   30 CONTINUE
C
      DO 50 K = 1,NROWS
   40 THETA(K,N) = -U0(K` * THETA(K,N)
      THS(K,N) = 0.00
   50 CONTINUE
      IF(NMAX .EQ. NWT) GO TO 80
      NWT1 = NWT + 1
      DO 70 I = NWT1,NMAX
   60 EPS(I,N) = 0.00
      BETA(I,N) = 0.00
   70 CONTINUE
   80 CONTINUE
      RETURN
      END

C     OVERLAY ALPHA
C     INSERT STAGE1,SGMAIN,INPTS,INPUTJ,XLETR1,XLETR2,NORM1,BOXS
C     INSERT INCASE,BEECEE,OUT1,INCOMP,BLOWIN,BOXJ,TANS
C     INSERT FCASE1,SG1
C     OVERLAY ALPHA
C     INSERT STAGE2
C     OVERLAY BETA
C     INSERT STG2D,DWNWSH,EVD1,EVD2,EVD3,EVD4,SHUFL1,SHUFL2,HINGE
C     INSERT COLUM1,COLUM2,PREP
C     OVERLAY BETA
C     INSERT STG2S,MATRIX,SAVE,GETT,BAKSUB,SOLV2
C     OVERLAY ALPHA
C     INSERT STAGE3,STG3FC,STG3FS,STG3FT,STG3C,EXPLE,EXPH1,EXPH2
C     INSERT SLOAD,SLOADX,SLOADG,TLOAD,TLOADD,TLOADX,TREFT2,TABLE
C     INSERT FUNDER,COMDER,SUMIT1,SUMIT2,STABLE
C     INSERT LOAD1,LOAD2,LOAD3,LOAD4,LOAD5,LOAD6
C     OVERLAY ALPHA
C     INSERT STAGE4,OUT2,SAVECP,BCPICH,BCROLL,BCYAW
C
C** ONR SAMPLE CASE ***  RECTANGULAR WING   CMU = 1   WITH STABILITY DER
C4.500     4.500     1.000     0.250     0.250
C4 3 0 0 0 2 1 1
C0.9750    0.88750   0.68750   0.2750
C1 1 2 1
C5 6
C0.000     0.100     0.200     0.500     0.900
C0.000     0.100     0.200     0.500     0.800     0.900
C4.500     0.000     1.000
C1 1 1 1
C4
C1.000     1.100     1.500     3.000
C0 0 1 0 0
```

```
C1.000     1.000     1.000     1.000
C0 0 0 0 1
C0 0 1 0
C0.9000   0 1.000
C1 0.00  2 10.00 3 10.00
C9
C1.000     1.000     1.000     1.000
C9
```

## PROGRAM JETFLAPIN LISTING

```
      PROGRAM JETFLAPIN
C
C     *** JETFLAPIN INPUT PROGRAM DEVELOPED BY J.A. CAMPBELL (AUG88)      ***
C     *** PROGRAM DESIGNED TO RUN UNDER FORTRAN 77 ON THE MICROVAX/2000 ***
C         FINAL UPDATES MADE 14 SEP 88 - (JAC)
C
C     THIS PROGRAM IS USED INTERACTIVELY TO PRODUCE AN INPUT FILE FOR THE
C     EVD JET WING COMPUTER PROGRAM, JETFLAP.  THE JETFLAP PROGRAM CALLS
C     THE FILE CREATED BY THIS PROGRAM AND WILL PROVIDE THE FOLLOWING
C     FOR WINGS OF ARBITRARY PLANFORM -
C
C             1.  SPANWISE AND CHORDWISE LOADING
C             2.  SPANWISE VARIATION OF INDUCED DRAG
C             3.  A CAPABILITY TO INVESTIGATE THE EFFECTS OF -
C                 A.  PART SPAN FLAPS
C                 B.  PART SPAN BLOWING
C                 C.  ROLLING, YAWING, PITCHING AND SIDESLIP
C             4.  TOTAL LIFT AND INDUCED DRAG (TREFFTZ PLANE METHOD),
C                 PITCHING, YAWING AND ROLLING MOMENTS, ETC.
C
C     COMPLETE DOCUMENTATION OF THE EVD JET WING LIFTING SURFACE THEORY
C     AND THE ASSOCIATED COMPUTER PROGRAM ARE CONTAINED IN DOUGLAS REPORT
C
C               J5519 -- A THEORETICAL METHOD FOR CALCULATING THE
C          AERODYNAMIC CHARACTERISTICS OF ARBITRARY JET FLAP WINGS
C
C                                VOLUME I
C               THE ELEMENTARY VORTEX DISTRIBUTION JET-WING
C                        LIFTING SURFACE THEORY
C
C                                VOLUME II
C             EVD JET-WING COMPUTER PROGRAM USERS MANUAL
C
C*****************************************************************
      INTEGER*4 LUN
      INTEGER*2 INFILE_SIZE,IOFILE_SIZE
      INTEGER    STATUS,NANS
      CHARACTER*20 INFILE,OUTFILE
      CHARACTER*4  CHECK
      LOGICAL    EXIST
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/LUKE/ TITLE(20)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1         D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TWIST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1         XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/SOLV1/B(600,10)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/DERIV/U0(40),CLQ,CMQ,CMQMC
      COMMON/INDAT/LUN
      DATA CHECK/'9    '/
      DATA LUN/7/
C------------------------------------------------------------------
C     CALL LIBRARY ROUTINE TO CLEAR THE SCREEN, THEN PRINT HEADER
C     TITLE PAGE AND INSTRUCTIONS
C------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, ' PROGRAM JETFLAPIN : VERSION 1 : 6 AUGUST 88 '
      PRINT *
      PRINT *, ' THIS PROGRAM DEVELOPS THE INPUT FILE REQUIRED BY'
      PRINT *, ' THE EVD JET-WING COMPUTER PROGRAM, JETFLAP. '
      PRINT *
      PRINT *, ' DOUGLAS AIRCRAFT COMPANY CREATED PROGRAM JETFLAP:'
      PRINT *, ' THE ELEMENTARY VORTEX DISTRIBUTION (EVD) JET-WING'
      PRINT *, ' COMPUTER PROGRAM FOR DETERMINING THE AERODYNAMIC'
      PRINT *, ' CHARACTERISTICS OF ARBITRARY JET FLAPPED WINGS'
      PRINT *
      PRINT *, ' A SIGNIFICANT AMOUNT OF INFORMATION REGARDING'
      PRINT *, ' YOUR WING PLANFORM IS REQUIRED BY THIS PROGRAM.'
      PRINT *, ' IF YOU HAVE NOT READ THE USERS MANUAL, YOU ARE'
      PRINT *, ' ENCOURAGED TO ANSWER NO TO THE FOLLOWING QUESTION'
      PRINT *, ' AND RETURN WITH YOUR PREPARED PLANFORM DATA.'
      PRINT *
C
    1 WRITE (6,1241)
      CALL QUERY (NANS)
      IF (NANS .EQ. 1) THEN
        GO TO 2
      ELSE IF(NANS .EQ. 2)THEN
        GO TO 110
      ELSE
        WRITE (6,1242)
        GO TO 1
      END IF
 6410 FORMAT (1X,'==================================================
     1=========================')
 6440 FORMAT (//,8X,'ENTER DATA FOR THE JETFLAP PROGRAM IN FREE FORMAT.
     1',/,8X,'AFTER EACH QUESTION THE FORMAT IS GIVEN: (R) - REAL,'
     2,2X,'(I) - INTEGER.',/,8X,'EXAMPLE: (R,R) INPUT 2.9,6.789 ',
```

```
     3'OR (I) INPUT 5',//)
1241  FORMAT (1X,' DO YOU WISH TO RUN THIS PROGRAM? 1 = YES;2 = NO')
1242  FORMAT (1X,'     INVALID RESPONSE - REENTER')
C-----------------------------------------------------------------------
C FOLLOWING LINES OPEN THE INPUT FILE TO BE CREATED
C-----------------------------------------------------------------------
    2 CONTINUE
      CALL CLRSCRN
      PRINT *
      PRINT *
      PRINT *, ' (ENTER 999 TO EXIT.)'
      STATUS = LIB$GET_INPUT (OUTFILE,                   ! The OUTPUT file
     2 ' ENTER NAME OF OUTPUT FILE TO CREATE:  ',        ! Prompt
     2                   IOFILE_SIZE)                     ! Filename size
C  CHECK TO SEE IF THE FILE EXISTS BEFORE CREATING IT
      IF (OUTFILE .EQ. '999') GO TO 1
      INQUIRE (FILE = OUTFILE (1:IOFILE_SIZE), EXIST = EXIST)
      IF ( EXIST ) THEN
         PRINT *
         PRINT *, ' THAT FILE ALREADY EXISTS.'
         WRITE (6,1243)
         PRINT *, ' (OR ENTER 999 TO RETURN TO EXIT OPTION).'
         PRINT *
    3    CALL QUERY (NANS)
      ELSE
         GO TO 4
      END IF
      IF (NANS .EQ. 1) THEN
         GO TO 4
      ELSE IF(NANS .EQ. 2)THEN
         GO TO 2
      ELSE IF(NANS .EQ. 999)THEN
         GO TO 1
      ELSE
         WRITE (6,1242)
         GO TO 3
      END IF
1243  FORMAT (1X,' DO YOU WISH TO OVERWRITE THIS FILE? 1 = YES;2 = NO')
C OPEN FILE THAT BECOMES INPUT FILE FOR PROGRAM JETFLAP
    4 OPEN (UNIT=LUN,
     2        FILE= OUTFILE,
     2        ORGANIZATION= 'SEQUENTIAL',
     2        ACCESS= 'SEQUENTIAL',
     2        RECORDTYPE= 'VARIABLE',
     2        FORM= 'FORMATTED',
     2        STATUS= 'UNKNOWN')
C-----------------------------------------------------------------------
C  INFORM USER OF DESIRED INPUT FORMATS AND ENTER FIRST LINE INPUT DATA
C  FIRST LINE INPUT DATA--THE PROBLEM TITLE FOR THIS CASE
C-----------------------------------------------------------------------
  8   CALL CLRSCRN
      WRITE (6,6410)
      WRITE (6,6440)
      WRITE (6,6410)
      WRITE (6,6450)
      WRITE (6,6460)
      READ (5,1000, END=100) TITLE
6450  FORMAT (1X,'************************** JETFLAP INPUT PARAMETERS ',
     1'***************************',/)
6460  FORMAT (1X,28H==> ENTER THE PROBLEM TITLE:,/,5X,20H(80 LETTERS MAX
     1IMUM))
1000  FORMAT (20A4)
1001  FORMAT (1X,20A4)
C-----------------------------------------------------------------------
C  SUMMARY OF FIRST LINE OF INPUT DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,570)
570   FORMAT (1X,'SUMMARY OF FIRST LINE OF INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 10
         WRITE (6,571)
         WRITE (6,1001) TITLE
         WRITE (6,575)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 8
571   FORMAT (1X,7X,'THE TITLE CARD FOR THIS DATA IS:')
575   FORMAT (//,1X,' DO YOU WISH TO CHANGE FIRST LINE OF INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
 10   CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN,1000) TITLE
C-----------------------------------------------------------------------
C  SECOND LINE INPUT DATA--GENERAL PLANFORM PARAMETERS
C-----------------------------------------------------------------------
C  READ GENERAL GEOMETRY CONTROL DATA
      CALL CLRSCRN
      PRINT *, '==> ENTER THE WING AREA, IN UNITS OF SPAN**2.'
      PRINT *, '     IF SPAN IS IN FEET, ENTER AREA IN SQUARE FEET.(R)'
      READ (5,*) AREA
      PRINT *
      PRINT *, '==> ENTER THE WING SPAN SEEN BY THE FREESTREAM'
      PRINT *, '     VELOCITY. USE ANY DESIRED UNITS.(R)'
      READ (5,*) SPAN
      PRINT *
      PRINT *, '==> ENTER CREF, THE WING REFERENCE CHORD. THIS WILL BE'
      PRINT *, '     USED FOR NORMALIZING VARIOUS AERODYNAMIC COEFFICIENT
     +S.'
      PRINT *, '     USE THE SAME UNITS AS SPAN.  IF YOU ENTER ZERO,'
      PRINT *, '     THE MEAN AERODYNAMIC CHORD WILL BE USED.(R)'
```

```
      READ (5,*) CREF
      PRINT *
      PRINT *, '==> ENTER XMC, THE POINT ABOUT WHICH PITCHING MOMENTS W
     +ILL BE'
      PRINT *, '    TAKEN, MEASURED FROM THE WING APEX (ORIGIN).'
      PRINT *, '    USE THE SAME UNITS AS SPAN.(R)'
      READ (5,*) XMC
      PRINT *
      PRINT *, '==> ENTER XCG, THE WING CENTER OF GRAVITY LOCATION, MEAS
     +URED'
      PRINT *, '    FROM THE WING APEX (ORIGIN). THIS WILL BE USED AS TH
     +E'
      PRINT *, '    PITCHING AXIS FOR COMPUTING THE STABILITY DERIVATIVE
     +S'
      PRINT *, '    DUE TO PITCHING. SAME UNITS AS SPAN.(R)'
      PRINT *, '    NOTE: THIS VALUE IS REQUIRED IF IDERIV IS NON-ZERO.'
      PRINT *, '       IF STABILITY DERIVATIVES NOT REQUIRED, ENTER 0.'
      PRINT *
      READ (5,*) XCG
C-----------------------------------------------------------------------
C  SUMMARY OF SECOND LINE INPUT DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,580)
580   FORMAT (1X,'SUMMARY OF SECOND LINE OF INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 20
         WRITE (6,581)
         WRITE (6,582) AREA,SPAN,CREF,XMC,XCG
         WRITE (6,590)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 10
581   FORMAT (1X,5X,'AREA',7X,'SPAN',7X,'CREF',7X,'XMC',8X,'XCG')
582   FORMAT (1X,5(1X,F10.3))
590   FORMAT (//,1X,33HCHANGE SECOND LINE OF INPUT DATA?,
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
   20 CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN,40) AREA,SPAN,CREF,XMC,XCG
   40 FORMAT(5F10.4)
C-----------------------------------------------------------------------
C  THIRD LINE INPUT DATA--GENERAL CONTROL PARAMETERS (FLAGS)
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *, '==> ENTER NROWS, THE NUMBER OF SPANWISE SECTIONS THE WIN
     +G'
      PRINT *, '    IS DIVIDED INTO. REQUIREMENT:(3.LE.NROWS.LE.40).(I)'
      READ (5,*) NROWS
      PRINT *
      PRINT *,'==> ENTER NCASES, THE TOTAL NUMBER OF FUNDAMENTAL CASES.'
      PRINT *,'    NCASES MUST BE ONE MORE THAN THE NUMBER OF CASES FOR'
      PRINT *,'    WHICH DATA INPUT WILL BE GIVEN TO ALLOW FOR THE ANGLE
     +'
      PRINT *, '    OF ATTACK CASE. REQUIREMENT:(1.LE.NCASES.LE.10).(I)'
      READ (5,*) NCASES
      PRINT *
      PRINT *, '==> ENTER ISYMM, THE X-AXIS SYMMETRY INDICATOR FLAG.(I)'
      PRINT *, '    = 0  WING AND JET ARE SYMMETRIC.'
      PRINT *, '    > 0  WING OR JET ARE NON-SYMMETRIC.'
      PRINT *, '    < 0  WING AND JET ARE ANTI-SYMMETRIC.'
      READ (5,*) ISYMM
      PRINT *
      PRINT *, '==> ENTER IPRINT, THE PRINTED OUTPUT CONTROL FLAG.(I)'
      PRINT *, '    > 1  PRINT GEOMETRY DETAILS AND TOTAL AERO COEFFS.'
      PRINT *, '    = 1  IN ADDITION, PRINT SPANWISE LOADING.'
      PRINT *, '    = 0  IN ADDITION, PRINT CHORDWISE LOADING.'
      PRINT *, '    < 0  IN ADDITION, PRINT ALL MATRICES, BACK SUBSTI-'
      PRINT *, '         TUTION CHECK AND OTHER DETAILS. (RESERVED FOR'
      PRINT *, '         TROUBLESHOOTING-VERY LARGE AMOUNTS OF OUTPUT.)'
      READ (5,*) IPRINT
      PRINT *
      PRINT *, '==> ENTER JETFLG, THE JET INDICATOR FLAG.(I)'
      PRINT *, ' *** WARNING: THIS VERSION NOT TESTED FOR JET INPUTS.**'
      PRINT *, '    = 0  THERE IS A JET SHEET.'
      PRINT *, '    = 1  THERE IS NO JET SHEET. NO JET INPUTS WILL BE RE
     +AD.'
      READ (5,*) JETFLG
      PRINT *
      PRINT *, '==> ENTER IGTYPE, THE WING PLANFORM GEOMETRY INDICATOR F
     +LAG.(I)'
      PRINT *, '    = 1  WING PLANFORM IS COMPLETELY ARBITRARY, AND SECT
     +IONAL'
      PRINT *, '         LEADING AND TRAILING EDGE COORDINATES WILL BE R
     +EAD'
      PRINT *, '         TO DEFINE THE PLANFORM.'
      PRINT *, '    = 2  WING PLANFORM IS TRAPEZOIDAL, AND SIMPLIFIED'
      PRINT *, '         PLANFORM INPUT WILL BE READ.'
      READ (5,*) IGTYPE
      PRINT *
      PRINT *, '==> ENTER IHINGE, THE HINGE EVD INDICATOR FLAG.(I)'
      PRINT *, '    = 0  REGULAR EVD ONLY WILL BE USED ON ALL HINGE ELEM
     +ENTS.'
      PRINT *, '    > 0  HINGE EVD WILL BE USED ON ALL HINGE ELEMENTS.
     +THIS'
      PRINT *, '         OPTION IS NOT PERMITTED FOR USE IN COMPUTING TH
     +E'
      PRINT *, '         DYNAMIC STABILITY DERIVATIVES (IDERIV>0).'
      READ (5,*) IHINGE
      PRINT *
      PRINT *,'==> ENTER IDERIV, THE DYNAMIC STABILITY DERIVATIVE FLAG.'
```

```
      PRINT *, '     = 0  A BASIC RUN WILL BE EXECUTED WITH NO STABILITY'
      PRINT *, '          DERIVATIVES COMPUTED.'
      PRINT *, '     > 0  A BASIC RUN WILL BE EXECUTED FOLLOWED BY A'
      PRINT *, '          DYNAMIC STABILITY DERIVATIVE RUN.'
      READ (5,*) IDERIV
      PRINT *
C-----------------------------------------------------------------------
C  SUMMARY OF THIRD LINE INPUT DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,600)
600   FORMAT (1X,'SUMMARY OF THIRD LINE OF INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
         CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 30
         WRITE (6,601)
         WRITE (6,602) NROWS,NCASES,ISYMM,IPRINT
         WRITE (6,603)
         WRITE (6,602) JETFLG,IGTYPE,IHINGE,IDERIV
         WRITE (6,604)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 20
601   FORMAT (1X,'NROWS',2X,'NCASES',1X,'ISYMM',2X,'IPRINT')
602   FORMAT (1X,4(2X,I2,2X),/)
603   FORMAT (1X,'JETFLG',1X,'IGTYPE',1X,'IHINGE',1X,'IDERIV')
604   FORMAT (//,1X,32HCHANGE THIRD LINE OF INPUT DATA?,
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
   30 CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN,41) NROWS,NCASES,ISYMM,IPRINT,
     1              JETFLG,IGTYPE,IHINGE,IDERIV
   41 FORMAT(10I2)
C-----------------------------------------------------------------------
      ARE = AREA
      SPA = SPAN
      CRE = CREF
      XM  = XMC
      XC  = XCG
      NRO = NROWS
      NC  = NCASES
      ISY = ISYMM
      IPR = IPRINT
      JET = JETFLG
      IGT = IGTYPE
      IHI = IHINGE
C
C  DETERMINE WHICH TYPE OF RUN IS DESIRED
      IF(IDERIV .NE. 0) GO TO 60
C
C  A REGULAR RUN WILL BE EXECUTED
   50 CALL APPLY1
      GO TO ( 60 , 70 , 100 , 120 ), IR
C
C  A STABILITY DERIVATIVE RUN WILL BE EXECUTED
   60 CALL APPLY2
      IF(IR .EQ. 2) GO TO 120
C
C***********************************************************************
C
C  PRINT COMPLETION MESSAGE FOR THIS RUN AND GO BACK TO BEGIN A NEW RUN
   70 WRITE(6, 80 )
   80 FORMAT(1H0///,32X,10(5H*****), 3H***/ 32X,
     1        53H*  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  */
     2        32X,10(5H*****),3H***)
C PUT IN AN OPTION TO DO ANOTHER RUN OR PRINT A '9' CARD AND QUIT.***
C
C       READ(5, 20, END=100) TITLE
C   90 IF(TITLE(1) .EQ. CHECK) GO TO 10
C       GO TO 30
C
C
      PRINT *, '==> DO YOU WISH TO ENTER ANOTHER SET OF DATA? (Y OR N)'
   90 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
        GO TO 4
      ELSE IF (ANS.EQ.'N') THEN
        WRITE(LUN,1010) CHECK
      ELSE
        PRINT *, '  INVALID RESPONSE - REENTER.'
        GO TO 90
      END IF
      PRINT *
 1010 FORMAT(A4)
C
C  PRINT COMPLETION MESSAGE AND STOP EXECUTION
  100 WRITE(6, 80 )
  110 STOP
C
C***********************************************************************
C  A FATAL ERROR HAS OCCURED.  PRINT FINAL MESSAGE AND STOP EXECUTION.
C***********************************************************************
  120 WRITE(6, 130 )
  130 FORMAT( 1H0///62X,2(4H****)/31X,11(5H*****)/
     1   31X,55H*  THE  PROGRAM  HAS  REACHED  ABNORMAL  TERMINATION  */
     2   31X,11(5H*****)/62X,2(4H****))
  140 STOP
C***********************************************************************
      END
      SUBROUTINE CLRSCRN
C
```

```
C  LIBRARY ROUTINE TO CLEAR THE SCREEN.
C
      ISTAT = LIB$ERASE_PAGE (1,1)
      RETURN
C*****************************************************************
      END
      SUBROUTINE QUERY(NANS)
C
C  ROUTINE TO TRAP ERRORS CAUSED BY IMPROPER RESPONSES TO QUESTIONS.
C  THE COMPUTER GENERATES AND ERROR WHEN A CHARACTER IS SUPPLIED TO
C  A QUESTION EXPECTING AN INTEGER OR REAL VALUE.
C
      NOTEST=0
    1 CONTINUE
      IF (NOTEST .GT. 0) THEN
          PRINT *, '  CHARACTER VALUES ARE NOT VALID.'
          PRINT *, '  PLEASE ENTER A VALUE OF 1 OR 2.'
      END IF
      NOTEST = NOTEST + 1
      READ (5,*,ERR=1)NANS
      RETURN
C*****************************************************************
      END
      SUBROUTINE APPLY1
C
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF INPUT FOR REGULAR CASES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C  DECIDE WHETHER OR NOT THERE IS AN ALPHA CASE
      NOALFA = 1
      IR = 1
      LOGIC = 1
      IF(ISYMM .LT. 0) NOALFA = 0
C
C  INITIALIZE AND INCREMENT THE CMU CASE CONTROL COUNTER
   10 NEWCMU = 0
   20 NEWCMU = NEWCMU + 1
C
C  EXECUTE THE PROBLEM FORMATION STAGE
   30 CALL STAGE1
      GO TO ( 40 , 60 , 70 , 80 ), IR
   40 CONTINUE
C
C  THE PROGRAM HAS BEEN EXECUTED SUCESSFULLY
C  GO BACK AND DO A NEW CMU CASE
      IF(JETFLG .NE. 0) GO TO 60
      GO TO 20
C
C*****************************************************************
C  THIS RUN HAS BEEN COMPLETED.  RETURN TO START A NEW RUN.
   60 IR = 2
      RETURN
C  THIS RUN HAS BEEN COMPLETED.  NO FURTHER RUNS FOLLOW.
   70 IR = 3
      RETURN
C  A FATAL ERROR HAS OCCURED.  RETURN AND QUIT.
   80 IR = 4
      RETURN
C*****************************************************************
      END
      SUBROUTINE APPLY2
C
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF INPUT FOR
C  STABILITY DERIVATIVES
C
C
C
C  CHECK ON STATUS OF CONTROL FLAGS
   10 IHINGE = 0
      NOALFA =1
      NEWCMU = 1
      IF(ISYMM .GE. 0) GO TO 30
      ISYMM = 0
      WRITE(6, 20 )
   20 FORMAT(1H0///16X,41HTHE ISYMM FLAG INDICATED AN ANTI-SYMETRIC,
     1     48H CASE.  HOWEVER, IT WILL BE TREATED AS SYMETRIC.)
C
C  EXCECUTE THE FIRST RUN
C
C  FORMULATE THE PROBLEM AS USUAL
   30 CALL STAGE1
      GO TO ( 40, 110, 100, 110), IR
   40 LOGIC = 1
C
C       WRITE(6, 70  )
   70 FORMAT( 1H1//////////// 37X,11(4H****),2H** /
     1          37X,46H*  SECOND RUN FOR STABILITY DERIVATIVE CASE  * /
     2          37X,11(4H****),2H**)
C  IF THIS IS A SYMETRIC WING, SWITCH IT TO ANTI-SYMETRIC FOR RUN 2
   80 IF(ISYMM .EQ. 0) ISYMM = -1
C
C*****************************************************************
C  THIS IS THE END OF THE LINE
  100 IR = 1
      RETURN
C  THE FOLLOWING LINE SHOULD NOT BE REACHED.  INCLUDED FOR CONTINUITY.
C  A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.
  110 IR = 2
      RETURN
C*****************************************************************
      END
```

```
      SUBROUTINE STAGE1
C
C  THIS SUBROUTINE READS THE GENERAL GEOMETRY PARAMETERS AND FLAGS, AND
C  CONTROLS THE CALLING OF THE SPECIALIZED GEOMETRY SUBROUTINES
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR
C
C  CHECK WHETHER THIS IS THE FIRST CMU CASE
      IF(NEWCMU .GT. 1) GO TO 50
      IF((NROWS .GT. 40) .OR. (NROWS .LT. 3)) GO TO 80
C
C  SECTIONAL INPUT
   10 IF((IGTYPE .EQ. 1).OR.(IGTYPE .EQ. 2))CALL SGMAIN(NOALFA,IR)
      GO TO ( 20 , 40 , 100 ), IR
C
C  USER INPUT ERROR
C
C  PRINT ERROR MESSAGE BECAUSE IGTYPE HAS THE WRONG VALUE
   20 WRITE(6, 30 ) IGTYPE
   30 FORMAT(1H1//44X,32HTHE IGTYPE FLAG HAS THE VALUE OF,I2/
     1       44X,37HONLY THE VALUES 1 OR 2 ARE ACCEPTABLE//
     2       44X,15HPLEASE REENTER.)
      READ(5,*) IGTYPE
      GO TO 10
C
C  READ THE COMPOSITE CASE REQUIREMENTS
   40 CALL INCOMP(NCASES,IR)
      IF(IR .EQ. 2) GO TO 100
C
C  READ THE CMU DATA
   50 CALL BLOWIN(JETFLG,IR)
      GO TO ( 60 , 110 , 120 ), IR
   60 CALL BOXJ(NEWMAX,IR)
      IF(IR .EQ. 2) GO TO 50
C
C  RETURN NORMALLY TO THE CONTROL PROGRAM
   70 IR = 1
      GO TO 130
C
C  PRINT ERROR MESSAGE BECAUSE THE NROWS VALUE IS UNACCEPTABLE
   80 WRITE(6, 90 ) NROWS
   90 FORMAT(1H1/55X,7HNROWS =,I3)
C
C*******************************************************************
C  A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.
  100 IR = 4
      GO TO 130
C  RETURN TO MAIN AND BEGIN A COMPLETELY NEW RUN
  110 IR = 2
      GO TO 130
C  RETURN TO MAIN AND STOP THE EXECUTION
  120 IR = 3
  130 RETURN
C*******************************************************************
      END
      SUBROUTINE SGMAIN(NOALFA,IR)
C
C  THIS SUBROUTINE CONTROLS ALL GEOMETRY CALCULATIONS FOR THE
C  SECTIONAL GEOMETRY METHOD
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
C
C  READ THE WING PLANFORM GEOMETRY DATA
   10 CALL INPTS(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 1) CALL XLETR1(IR)
      IF(IR .EQ. 2) GO TO 100
      IF(IGTYPE .EQ. 2) CALL XLETR2
C
C  NORMALIZE THE WING PLANFORM GEOMETRY DATA
   20 CALL NORM1
C
C  READ THE JET SHEET GEOMETRY DATA
   30 CALL INPUTJ(IR)
      IF(IR .EQ. 2) GO TO 100
C
C  CONSTRUCT THE EVD ELEMENTS
   40 CALL BOXS(IR)
      IF(IR .EQ. 2) GO TO 100
C
C  CONSTRUCT THE SET OF FUNDAMENTAL GEOMETRIC CASES
C
      DO 90 N = 1,NCASES
      LCASE = N
C
C  READ THE GEOMETRY FOR THIS CASE
   50 CALL INCASE(LCASE,NOALFA)
C
C  PRINT THE GEOMETRY AND CONSTRUCTED CASE DATA IF DESIRED
      IF(LCASE .EQ. 1) WRITE(6, 70 )
   70 FORMAT(1H1)
C
      CALL CLRSCRN
      PRINT *
      PRINT *, '==> DO YOU WISH TO SEE THE CONSTRUCTED CASE DATA?'
      PRINT *, '      ENTER (Y OR N)'
   75 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
        GO TO 80
```

```
      ELSE IF (ANS.EQ.'N') THEN
         GO TO 90
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 75
      END IF
      PRINT *
 1010 FORMAT(A4)
C
   80 CALL OUT1(LCASE)
   90 CONTINUE
      IR = 2
      RETURN
C
C  AN ERROR HAS OCCURED.   RETURN ABNORMALLY TO STAGE1.
C
  100 IR = 3
      RETURN
C*****************************************************************
      END
      SUBROUTINE INPTS(IR)
C
C  THIS SUBROUTINE READS THE WING GEOMETRY DATA FROM THE KEYBOARD
C  FOR THE SECTIONAL GEOMETRY METHOD
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1       D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1       NWTYPE,NJTYPE
      COMMON/INDAT/LUN
      DIMENSION NI(10)
C
C  INPUT THE SECTIONAL PLANFORM DATA
   10 NWTYPE = 0
C-----------------------------------------------------------------
C  SECTION CENTERLINE LOCATION CARDS
C-----------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '    SECTION CENTERLINE LOCATION VALUES'
      PRINT *
      PRINT *, '==> ENTER Y, THE SPANWISE DISTANCE FROM THE CENTERLINE,'
      PRINT *, '    (X-AXIS), TO THE SECTION CENTERLINE, NORMALIZED BY'
      PRINT *,'    THE HALF-SPAN, SPAN/2. REQUIREMENT: (-1.0<Y<1.0).(R)'
      PRINT *
      PRINT *, '    BEGIN AT THE RIGHT WING TIP AND WORK TOWARD:'
      PRINT *, '     a) WING CENTERLINE FOR SYMETTRIC OR ANTISYMMETRIC W
     +INGS.'
      PRINT *, '     b) LEFT WING TIP FOR NON-SYMMETRIC WINGS.'
      PRINT *
      PRINT *, '    A MAXIMUM OF 40 SECTIONS IS ALLOWED.'
      PRINT *
      DO 15 K = 1,NROWS
      WRITE(6,12) K,NROWS
      READ(5, * ) Y(K)
   12 FORMAT(1X,' ENTER SECTION CENTERLINE ',I2,' OF ',I2,' SECTIONS.',
     +/)
      PRINT *
   15 CONTINUE
C-----------------------------------------------------------------
C  SUMMARY OF SECTION CENTERLINE INPUT DATA
C-----------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '  *** YOU ARE ENCOURAGED TO CHECK THIS DATA! *** '
      PRINT *
      PRINT *, '  IF THE SECTION CENTERLINE VALUES ARE NOT VALID'
      PRINT *, '  AN ERROR WILL BE DETECTED BY SUBROUTINE BOXS,'
      PRINT *, '  THE PROGRAM WILL TERMINATE, AND YOU WILL HAVE TO'
      PRINT *, '  REENTER ALL YOUR DATA.  THE CHOICE IS YOURS...'
      PRINT *
      WRITE (6,13)
   13 FORMAT (1X,'SUMMARY OF SECITON CENTERLINE INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 20
         WRITE (6,14)
         WRITE (6,19) (K, Y(K),K=1,NROWS)
         WRITE (6,16)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 10
   14 FORMAT (1X,7X,'THE SECTION CENTERLINE DATA IS:')
   16 FORMAT (//,1X,'DO YOU WISH TO CHANGE/REENTER THIS INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
   20 CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN, 18 ) (Y(K),K=1,NROWS)
   18 FORMAT(8F10.6)
   19 FORMAT(1X,5X,'SECTION =',I2,3X,'CENTERLINE =',F10.6)
C-----------------------------------------------------------------
C  WING SECTION TYPE CARDS
C-----------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '    WING SECTION TYPES'
      PRINT *
      PRINT *, '==> ENTER ICTYPE, THE TYPE NUMBER OF EACH WING SECTION.'
      PRINT *, '    THE ARRANGEMENT OF EVD ELEMENTS IN A ROW DETERMINES'
      PRINT *, '    THE WING ROW TYPE. ANY SECTIONS HAVING THE SAME NUMB
```

```
     +ER'
      PRINT *, '     OF ELEMENTS, ALL WITH THE SAME SPACING FROM THE SECT
     +ION'
      PRINT *, '     LEADING EDGE ARE OF THE SAME ICTYPE. BEGIN WITH A TY
     +PE'
      PRINT *,'    1 AND WORK IN SEQUENCE, 2,3,...(ASCENDING ORDER).(I)'
      PRINT *
      PRINT *, '     A MAXIMUM OF 10 SECTION TYPES IS ALLOWED.'
   23 PRINT *
      DO 24 K = 1,NROWS
      WRITE(6,22) K,NROWS
      READ(5, * ) ICTYPE(K)
      IF(ICTYPE(K) .GT. NWTYPE) NWTYPE = ICTYPE(K)
      IF(NWTYPE .GT. 8) THEN
         WRITE(6,21) NWTYPE
         PRINT *, '     A MAXIMUM OF 10 SECTION TYPES IS ALLOWED.'
         PRINT *, '          ***** W A R N I N G *****'
         PRINT *, '     YOU MAY ENTER ONLY ONE MORE DIFFERENT TYPE.'
      END IF
   21 FORMAT(1X,5X,26HNUMBER OF WING ROW TYPES =,I3)
   22 FORMAT(1X,'    ENTER SECTION TYPE FOR SECTION ',I2,' OF ',I2,' SECTI
     +ONS.',/)
   24 CONTINUE
C----------------------------------------------------------------------
C  SUMMARY OF WING SECTION TYPE INPUT DATA
C----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,26)
   26 FORMAT (1X,'SUMMARY OF WING SECTION TYPE INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 25
         WRITE (6,27)
         WRITE (6,29) (K,ICTYPE(K),K=1,NROWS)
         WRITE (6,16)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 23
   27 FORMAT (1X,7X,'THE WING SECTION TYPE DATA IS:')
   29 FORMAT(1X,5X,'SECTION =',I2,3X,'SECTION TYPE =',I2)
   25 CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN, 301 ) (ICTYPE(K),K=1,NROWS)
  301 FORMAT(40I2)
C----------------------------------------------------------------------
C  NUMBER OF CHORDWISE WING ELEMENTS CARD
C----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '     CHORDWISE WING ELEMENTS'
      PRINT *
      PRINT *, '==> ENTER NI, THE NUMBER OF CHORDWISE WING EVD ELEMENTS'
      PRINT *, '     FOR EACH WING SECTION TYPE. THE NUMBER OF ELEMENTS M
     +UST'
      PRINT *, '     BE ENTERED IN ASCENDING ORDER BY ICTYPE. THERE MAY B
     +E AS'
      PRINT *, '     FEW AS 2 ELEMENTS PER SECTION TYPE OR AS MANY AS 20.
     +(I)'
   33 PRINT *
      DO 30 N = 1,NWTYPE
      WRITE(6,32) N,NWTYPE
   28 READ(5, * ) NI(N)
      NIN = NI(N)
      IF((NIN .LT. 2) .OR. (NIN .GT. 20)) THEN
         WRITE(6,31) NIN
         PRINT *, '     A MINIMUM OF 2 AND A MAXIMUM OF 20'
         PRINT *, '        ELEMENTS ARE ALLOWED.'
         PRINT *, '          PLEASE REENTER'
         GO TO 28
      END IF
   31 FORMAT(1X,5X,36HNUMBER OF ELEMENTS IN THIS SECTION =,I3)
   32 FORMAT(1X,'    ENTER NUMBER OF EVD ELEMENTS FOR ICTYPE ',I2,' OF ',I
     +2,/)
   30 CONTINUE
C----------------------------------------------------------------------
C  SUMMARY OF CHORDWISE WING ELEMENTS INPUT DATA
C----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,36)
   36 FORMAT (1X,'SUMMARY OF CHORDWISE WING ELEMENTS INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 35
         WRITE (6,37)
         WRITE (6,39) (N,NI(N),N=1,NWTYPE)
         WRITE (6,16)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 33
   37 FORMAT (1X,7X,'THE CHORDWISE WING ELEMENTS DATA IS:')
   39 FORMAT(1X,5X,'SECTION =',I2,3X,'CHORDWISE ELEMENTS =',I2)
   35 CONTINUE
C  WRITE DATA TO FILE
      WRITE(LUN, 301 ) (NI(N),N=1,NWTYPE)
C
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
C----------------------------------------------------------------------
C  WING CHORDWISE ELEMENT COORDINATES CARD
C----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '     WING CHORDWISE ELEMENT COORDINATES'
      PRINT *
```

```
      PRINT *, '==> ENTER XBW, THE CHORDWISE COORDINATE OF EACH VORTEX P
     +OINT.'
      PRINT *, '     THE VORTEX POINT IS DEFINED AS THE LEADING EDGE FOR'
      PRINT*,'   LEADING EDGE EVD''S AND THE "PEAK" POINT FOR REGULAR,'
      PRINT *, '     HINGE AND JET EVD''S.'
      PRINT *, '     A SET OF COORDINATES IS REQUIRED FOR EACH WING SECTI
     +ON TYPE.'
      PRINT *, '     THE NUMBER OF OF COORDINATES WILL CORRESPOND TO THE
     +NUMBER'
      PRINT *, '     OF ELEMENTS ENTERED ON THE PREVIOUS CARD.'
      PRINT *, '     THE LEADING EDGE COORDINATE MUST BE 0.0 AND WILL AUT
     +OMATICALLY'
      PRINT *, '     BE ENTERED FOR YOU. THE LAST VALUE MUST BE LESS THAN
     + 1.0.(R)'
      PRINT *
      DO 50 N = 1,NWTYPE
      NIN = NI(N)
      XBW(1,N) = 0.0
         DO 45 L = 2,NIN
            WRITE(6,42) N
            WRITE(6,43) L,NIN
   46       READ(5, * ) XBW(L,N)
            XBWN = XBW(L,N)
            IF((XBWN .LT. 0.0) .OR. (XBWN .GE. 1.0)) THEN
               WRITE(6,41) XBWN
            PRINT *, '    COORDINATE VALUE MUST LIE BETWEEN 0.0 AND 1.0'
               PRINT *, '            PLEASE REENTER'
               GO TO 46
            END IF
   45    CONTINUE
        CALL CLRSCRN
        PRINT *
        PRINT *, '  *** LEADING EDGE VALUE OF 0.0 ENTERED FOR EVD',
     +  ' ELEMENT 1 ***'
   50 CONTINUE
   41 FORMAT(1X,5X,30HCHORDWISE ELEMENT COORDINATE =,F10.6)
   42 FORMAT(1X,'  FOR WING SECTION TYPE NUMBER ',I2)
   43 FORMAT(1X,'  ENTER CHORDWISE COORDINATE FOR EVD ELEMENT ',I2,' OF
     +',I2,/)
C-----------------------------------------------------------------------
C  SUMMARY OF CHORDWISE ELEMENT COORDINATE INPUT DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,47)
   47 FORMAT (1X,'SUMMARY OF ELEMENT COORDINATE INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 60
   54    CALL CLRSCRN
         WRITE (6,48) NWTYPE
         READ (5,*) NSEC
         WRITE (6,49) (L,XBW(L,NSEC),L=1,NI(NSEC))
         WRITE (6,16)
         CALL QUERY (NANS)
      IF (NANS.EQ.1) THEN
         N = NSEC
         NIN = NI(N)
         XBW(1,N) = 0.0
         DO 55 L = 2,NIN
            WRITE(6,42) NSEC
            WRITE(6,43) L,NIN
   56       READ(5, * ) XBW(L,N)
            XBWN = XBW(L,N)
            IF((XBWN .LT. 0.0) .OR. (XBWN .GE. 1.0)) THEN
               WRITE(6,41) XBWN
            PRINT *, '    COORDINATE VALUE MUST LIE BETWEEN 0.0 AND 1.0'
               PRINT *, '            PLEASE REENTER'
               GO TO 56
            END IF
   55    CONTINUE
         GO TO 50
      ELSE
         PRINT *, ' DO YOU WISH TO CHECK ANOTHER SECTION?'
         PRINT *, ' ==> ENTER 1 = YES; 2 = NO'
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 54
        CONTINUE
      END IF
   48 FORMAT (1X,7X,'WHICH SECTION TYPE DO YOU WANT TO LOOK AT?'
     1,/,1X,7X,'ENTER A VALUE BETWEEN 1 AND ',I2,'.')
   49 FORMAT (1X,5X,'ELEMENT NUMBER =',I2,3X,'COORDINATE =',F10.6)
   60 CONTINUE
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW, USED LATER
      DO 70  K = 1,NROWS
      ICK = ICTYPE(K)
      NW(K) = NI(ICK)
   70 CONTINUE
C  WRITE DATA TO FILE
      DO 80 N = 1,NWTYPE
      NIN = NI(N)
      WRITE(LUN, 18 ) (XBW(L,N),L=1,NIN)
   80 CONTINUE
C
      IR = 1
      RETURN
C***********************************************************************
      END
      SUBROUTINE INPUTJ(IR)
C
C  THIS SUBROUTINE READS THE JET ELEMENT GEOMETRY INPUT
```

```
C  THE NUMBER AND CHORDWISE SPACING OF THE JET ELEMENTS ARE READ
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1           NWTYPE,NJTYPE
      COMMON/INDAT/LUN
      DIMENSION NI(10)
C
C  READ THE TYPE OF DIVISION FOR EACH ROW
   10 NJTYPE = 0
      NROWSJ = 0
      IF(JETFLG .NE. 0) GO TO 90
C-----------------------------------------------------------------------
C  JET SECTION TYPE CARDS
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '    JET SECTION TYPE NUMBERS'
      PRINT *
      PRINT *, '    THIS IS VERY SIMILAR TO THE WING SECTION TYPE DATA'
      PRINT *, '    COMPLETED PREVIOUSLY.'
      PRINT *
      PRINT *, '    THE ARRANGEMENT OF JET ELEMENTS IN A SECTION DETERMI
     +NES'
      PRINT *, '    THE JET SECTION TYPE. ANY SECTIONS HAVING THE SAME N
     +UMBER'
      PRINT *, '    OF ELEMENTS, ALL WITH THE SAME SPACING WITH RESPECT'
     + , ' TO'
      PRINT *, '    THE WING SECTIONAL CHORD TO WHICH THEY ARE ATTACHED'
     + , ' ARE'
      PRINT *, '    OF THE SAME TYPE. BEGIN WITH A TYPE NUMBER OF 1 AND'
     + , ' WORK'
      PRINT *, '    IN SEQUENCE, 2,3,...(ASCENDING ORDER).'
      PRINT *
      PRINT *, '    A SECTION WITH NO JET HAS A TYPE OF 0 (ZERO).'
      PRINT *
      PRINT *, '    A MAXIMUM OF 10 JET SECTION TYPES IS ALLOWED. THERE'
      PRINT *, '    IS A REQUIREMENT THAT THE SECTIONS WITH JETS AND'
      PRINT *, '    WITHOUT JETS MUST BE IN GROUPS OF THREE OR MORE.'
      PRINT *
      PRINT *, '==> ENTER IJTYPE, THE TYPE NUMBER OF EACH JET SECTION.(I
     +)'
   20 PRINT *
      DO 25 K = 1,NROWS
      WRITE(6,32) K,NROWS
      READ(5, * ) IJTYPE(K)
      IF(IJTYPE(K) .GT. NJTYPE) NJTYPE = IJTYPE(K)
      IF(IJTYPE(K) .NE. 0) NROWSJ = NROWSJ + 1
      IF(NJTYPE .GT. 8) THEN
         WRITE(6,31) NJTYPE
         PRINT *, '    A MAXIMUM OF 10 JET SECTION TYPES IS ALLOWED.'
         PRINT *, '          ***** W A R N I N G *****'
         PRINT *, '    YOU MAY ENTER ONLY ONE MORE DIFFERENT TYPE.'
      END IF
   31 FORMAT(1X,5X,29HNUMBER OF JET SECTION TYPES =,I3)
   32 FORMAT(1X,' ENTER JET SECTION TYPE FOR SECTION ',I2,' OF ',I2,' S
     +ECTIONS.',/)
   25 CONTINUE
C
C  SET UP FOR ROW CONSISTENCY CHECK
C
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW
      DO 80  K = 1,NROWS
      NJ(K) = 0
      IF(IJTYPE(K) .EQ. 0) GO TO 80
      IJK = IJTYPE(K)
   70 NJ(K) = NI(IJK)
   80 CONTINUE
C  CHECK FOR ROW CONSISTENCY ON EITHER SIDE OF JET
      ICOUNT = 1
      IF(NJ(1) .EQ. 0)ITEST = 0
      IF(NJ(1) .GT. 0)ITEST = 1
      DO 150 K = 2,NROWS
      IF(NJ(K) .EQ. 0)ICOMP = 0
      IF(NJ(K) .GT. 0)ICOMP = 1
      IF(ICOMP .EQ. ITEST)GO TO 160
      IF(ICOUNT .LT. 3)GO TO 170
      ICOUNT = 1
      IF(NJ(K) .EQ. 0)ITEST = 0
      IF(NJ(K) .GT. 0)ITEST = 1
      GO TO 150
  160 ICOUNT = ICOUNT + 1
  150 CONTINUE
      IF(ICOUNT .LT. 3) GO TO 170
      GO TO 190
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND ENTER DATA AGAIN.
  170 PRINT *, '   ROW CONTINUITY RULE FAILURE!'
      PRINT *, '   REVIEW YOUR JET SECTION DATA. JET SECTIONS MUST BE'
      PRINT *, '   IN GROUPS OF 3 OR MORE AND THERE MUST BE AT LEAST 3'
      PRINT *,'  UNBLOWN WING SECTIONS INBOARD OR OUTBOARD OF ANY JET.'
      GO TO 20
  190 CONTINUE
C
C  SUMMARY GOES HERE
C
C-----------------------------------------------------------------------
C  NUMBER OF CHORDWISE JET ELEMENTS CARD
C-----------------------------------------------------------------------
      CALL CLRSCRN
```

```
      PRINT *
      PRINT *, '    CHORDWISE JET ELEMENTS'
      PRINT *
      PRINT *, '==> ENTER NI, THE NUMBER OF CHORDWISE JET EVD ELEMENTS'
      PRINT *, '    FOR EACH JET SECTION TYPE. THE NUMBER OF ELEMENTS ',
     +'MUST'
      PRINT *, '    BE ENTERED IN ASCENDING ORDER BY IJTYPE. THERE MAY',
     +' BE AS'
      PRINT *, '    FEW AS 2 ELEMENTS PER SECTION TYPE OR AS MANY AS 10'
     +,'.(I)'
      PRINT *
C  READ THE NUMBER OF CHORDWISE DIVISIONS (ELEMENTS) IN EACH ROW TYPE
      DO 30 N = 1,NJTYPE
      WRITE(6,22) N,NJTYPE
  28  READ(5, * ) NI(N)
      NIN = NI(N)
      IF((NIN .LT. 2) .OR. (NIN .GT. 10)) THEN
         WRITE(6,21) NIN
         PRINT *, '      A MINIMUM OF 2 AND A MAXIMUM OF 10'
         PRINT *, '          ELEMENTS ARE ALLOWED.'
         PRINT *, '            PLEASE REENTER'
         GO TO 28
      END IF
   21 FORMAT(1X,5X,40HNUMBER OF JET ELEMENTS IN THIS SECTION =,I3)
   22 FORMAT(1X,'  ENTER NUMBER OF JET ELEMENTS FOR IJTYPE ',I2,' OF ',I
     +2,/)
   30 CONTINUE
C
C  SUMMARY GOES HERE
C
C
C-----------------------------------------------------------------------
C  JET CHORDWISE ELEMENT COORDINATES CARD
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '    JET CHORDWISE ELEMENT COORDINATES'
      PRINT *
      PRINT *, '    A SET OF COORDINATES IS REQUIRED FOR EACH JET ',
     +'SECTION TYPE.'
      PRINT *, '    THE NUMBER OF OF COORDINATES WILL CORRESPOND TO ',
     +'THE NUMBER'
      PRINT *, '    OF ELEMENTS ENTERED ON THE PREVIOUS CARD.'
      PRINT *, '    THE FIRST PEAK POINT FOR EACH JET SECTION OCCURS ',
     +'AT THE'
      PRINT *, '    TRAILING EDGE. ITS COORDINATE MUST BE 1.0 AND WILL',
     +' AUTOMATICALLY'
      PRINT *, '    BE ENTERED FOR YOU. THERE IS NO MAXIMUM VALUE.'
      PRINT *
      PRINT *, '==> ENTER XBJ, THE CHORDWISE COORDINATE OF EACH VORTEX',
     +' POINT.'
      PRINT *, '    THE VORTEX POINT IS DEFINED AS THE "PEAK" POINT ',
     +'FOR JET EVD''S.(R)'
      PRINT *
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE
      DO 50 N = 1,NJTYPE
      NIN = NI(N)
      XBJ(1,N) = 1.0
         DO 45 L = 2,NIN
            WRITE(6,42) NJTYPE
            WRITE(6,43) L,NIN
            PRINT *, '      NOTE: THIS IS WITH RESPECT TO THE CHORD OF ',
     +               'THIS SECTION.'
   46       READ(5, * ) XBJ(L,N)
            XBJN = XBJ(L,N)
            IF(XBJN .LE. 1.0) THEN
               WRITE(6,41) XBJN
               PRINT *, '    COORDINATE VALUE MUST BE GREATER THAN 1.0'
               PRINT *, '            PLEASE REENTER'
               GO TO 46
            END IF
           PRINT *
   45     CONTINUE
   50 CONTINUE
   41 FORMAT(1X,5X,30HCHORDWISE ELEMENT COORDINATE =,F10.6)
   42 FORMAT(1X,'  FOR JET SECTION TYPE NUMBER ',I2)
   43 FORMAT(1X,'  ENTER CHORDWISE COORDINATE FOR JET EVD ELEMENT ',I2,
     +' OF ',I2,/)
C
C  SUMMARY GOES HERE
C
      IR = 1
      RETURN
C
C  THERE IS NO JET FOR THIS RUN
   90 DO 100 K = 1,NROWS
      IJTYPE(K) = 0
      NJ(K) = 0
  100 CONTINUE
      IR = 1
      RETURN
C***********************************************************************
      END
      SUBROUTINE XLETR1(IR)
C
C  THIS SUBROUTINE READS THE LEADING AND TRAILING EDGE COORDINATES AT
C  SPANWISE STATIONS CONNECTED BY STRAIGHT LEADING AND TRAILING EDGES.
C  THE MAIN PROGRAM INTERPOLATES TO GET COORDINATES FOR INTERMEDIATE
C  SECTIONS
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
```

```
     1                 D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/INDAT/LUN
      DIMENSION YP(40),XLE(40),XTR(40)
C
C-----------------------------------------------------------------------
C   LEADING AND TRAILING EDGE COORDINATES
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '     LEADING AND TRAILING EDGE COORDINATES'
      PRINT *, '     FOR A WING OF ARBITRARY PLANFORM'
      PRINT *
      PRINT *, '==> ENTER AS A MINIMUM THE COORDINATES FOR THE TIP AND',
     +         '  ROOT SECTIONS.'
      PRINT *, '     COORDINATES ARE ALSO REQUIRED FOR SECTIONS WHICH ',
     +        'DEFINE A'
      PRINT *, '     BREAK IN THE LEADING OR TRAILING EDGES.'
      PRINT *, '     THE COORDINATES REFER TO THE CHORDWISE DISTANCE, ',
     +         'MEASURED AT'
      PRINT *, '     THE SECTION CENTERLINE, FROM THE Y-AXIS TO THE ',
     +         'RESPECTIVE EDGE'
      PRINT *, '     IN UNITS OF SPAN.'
      PRINT *, '     THE PROGRAM ASSUMES A STRAIGHT EDGE EXISTS BETWEEN',
     +        ' SECTIONS'
      PRINT *, '     ENTERED HERE AND WILL INTERPOLATE BETWEEN THE INPUT'
     +,        ' VALUES.'
      PRINT *, '     THE SECTION CENTERLINE COORDINATE IS AUTOMATICALLY',
     +        ' WRITTEN'
      PRINT *, '     TO THE DATA FILE FROM YOUR PREVIOUS INPUT.(R,R)'
      PRINT *
    5 NX = 0
C   READ NUMBER OF SECTIONS TO INPUT
    9 PRINT *, '     HOW MANY WING SECTIONS WILL YOU BE ENTERING',
     +        ' COORDINATES FOR?'
   10 READ(5, * ) NSECT
      IF (NSECT. GT. NROWS) THEN
        WRITE(6,11) NROWS
        PRINT *, '          PLEASE REENTER'
        GO TO 10
      END IF
C   CHANGE TO CORRECT NSECT IF IMPROPER VALUE ENTERED
      PRINT *
      WRITE(6,31) NSECT
      WRITE (6,16)
      CALL QUERY (NANS)
      IF (NANS.EQ.1) GO TO 9
C   READ XLEAD AND XTRAIL
      PRINT *, '     BEGIN AT TIP AND WORK IN.  TIP SECTION = 1.'
      PRINT *
      DO 30 N = 1,NSECT
      IF (N .NE. 1) CALL CLRSCRN
      WRITE(6,46) N,NSECT
   20 READ(5, * ) I
C RETRIEVE AND PRINT CENTERLINE COORDINATE DATA
      YP(N) = Y(I)
      WRITE(6,42) I
      WRITE(6,43) YP(N)
      PRINT *
      WRITE(6,44) I
      READ(5, * ) XLE(N)
      PRINT *
      WRITE(6,45) I
      READ(5, * ) XTR(N)
      PRINT *
   30 CONTINUE
   11 FORMAT(1X,5X,'THE NUMBER OF SECTIONS MUST NOT BE MORE THAN ',I2,/)
   31 FORMAT(1X,5X,'THE NUMBER OF SECTIONS YOU WILL BE ENTERING DATA'
     1,'  FOR IS ',I2)
   41 FORMAT(1X,5X,30HCHORDWISE ELEMENT COORDINATE =,F10.6)
   42 FORMAT(1X,'   FOR SECTION (ROW) NUMBER ',I2)
   43 FORMAT(1X,'   SECTION CENTERLINE COORDINATE = ',F10.6)
   44 FORMAT(1X,'   ENTER THE LEADING EDGE COORDINATE FOR SECTION ',I2)
   45 FORMAT(1X,'   ENTER THE TRAILING EDGE COORDINATE FOR SECTION ',I2)
   46 FORMAT(1X,'   ENTER THE WING SECTION NUMBER ASSOCIATED WITH '
     1,'COORDINATE SET ',I2,' OF ',I2,/)
C
C-----------------------------------------------------------------------
C   SUMMARY OF LEADING AND TRAILING EDGE COORDINATES DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,47)
 47   FORMAT (1X,'SUMMARY OF LEADING/TRAILING EDGE COORDINATE DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 60
          WRITE (6,48)
          WRITE (6,49)
          WRITE (6,52) (YP(N),XLE(N),XTR(N),N=1,NSECT)
          WRITE (6,16)
          CALL QUERY (NANS)
          IF (NANS.EQ.1) GO TO 5
   48 FORMAT (1X,7X,'THE COORDINATE DATA IS:',/)
   16 FORMAT (//,1X,'DO YOU WISH TO CHANGE/REENTER THIS INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
   49 FORMAT(1X,5X,'CENTERLINE',5X,'LEADING EDGE',3X,'TRAILING EDGE',/)
   52 FORMAT(3(5X,F10.6))
   60 CONTINUE
C   WRITE DATA TO FILE
      DO 70 N = 1,NSECT
      WRITE(LUN, 101 )YP(N),XLE(N),XTR(N)
```

```
  101 FORMAT(3F10.6)
   70 CONTINUE
C
C   OUTPUT A 9 CARD AFTER NSECT SETS OF COORDINATES HAVE BEEN INPUT
      WRITE(LUN, 102 )
  102 FORMAT('9   ')
C
      IR = 1
      RETURN
C*****************************************************************
      END
      SUBROUTINE XLETR2
C
C   THIS SUBROUTINE READS THE FUNDAMENTAL PLANFORM PARAMETERS FOR A
C   TRAPEZOIDAL WING.  NOTE THAT THE PLANFORM OUTLINE MUST BE SYMETRIC.
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/INDAT/LUN
C
C-----------------------------------------------------------------
C   TRAPEZOIDAL WING PLANFORM PARAMETERS
C-----------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '     TRAPEZOIDAL WING PLANFORM PARAMETERS'
      PRINT *, '     NOTE: PLANFORM MUST BE SYMMETRIC'
      PRINT *
C CALCULATE ASPECT RATIO FROM PREVIOUSLY SUPPLIED DATA
      ARATIO = SPAN * SPAN / AREA
      PRINT *, '==> THE CALCULATED WING ASPECT RATIO, ARATIO =', ARATIO
      GO TO 15
   10 PRINT *
C  READ THE FUNDAMENTAL PLANFORM PARAMETERS
      PRINT *, '==> ENTER THE WING ASPECT RATIO, ARATIO (R).'
      READ (5,*) ARATIO
   15 PRINT *
      PRINT *, '==> ENTER SWEEP, THE SWEEP ANGLE OF THE QUARTER-CHORD'
      PRINT *, '    LINE, IN DEGREES.(R)'
      READ (5,*) SWEEP
      PRINT *
      PRINT *, '==> ENTER TR, THE WING TAPER RATIO. THIS IS DEFINED AS'
      PRINT *, '    THE CHORD AT THE WING TIP DIVIDED BY THE CHORD AT'
      PRINT *, '    THE AXIS OF SYMMETRY, THE WING ROOT.(R)'
      READ (5,*) TR
      PRINT *
C
C-----------------------------------------------------------------
C   SUMMARY OF TRAPEZOIDAL WING PLANFORM PARAMETERS INPUT DATA
C-----------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,11)
   11 FORMAT (1X,'SUMMARY OF TRAPEZOIDAL PLANFORM PARAMETERS DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
      CALL QUERY (NANS)
      IF (NANS.GE.2) GO TO 20
         PRINT *
         WRITE (6,12) ARATIO,SWEEP,TR
         WRITE (6,16)
         CALL QUERY (NANS)
         IF (NANS.EQ.1) GO TO 10
   12 FORMAT (1X,'ASPECT RATIO =',F10.6,3X,'SWEEP =',F10.6,
     13X,'TAPER RATIO =',F10.6,/)
   16 FORMAT (//,1X,'DO YOU WISH TO CHANGE/REENTER THIS INPUT DATA?',
     1/,1X,25H==> ENTER 1 = YES; 2 = NO)
   20 CONTINUE
C  WRITE TO DATA FILE
      WRITE(LUN, 100 ) ARATIO,SWEEP,TR
  100 FORMAT(3F10.6)
C
C   PROCESS VALUES FOR USE BY CHECKING ROUTINES
C
C   COMPUTE THE GENERAL PLANFORM CHARACTERISTICS
      B2 = SPAN / 2.00
      SW = SWEEP / 57.295779
      CROOT = 2.0 * SPAN / ((1.0+TR)*ARATIO)
      AREA = (1.0+TR) * CROOT * B2
      XLB2 = 0.250 * (1.0-TR) * CROOT + B2 * TAN(SW)
      CMAC = 2.0 * CROOT * (1.0 + TR + TR*TR) / (3.0*(1.0+TR))
      IF(CREF .EQ. 0.0) CREF = CMAC
      CBAR=AREA/SPAN
C
C   COMPUTE THE LEADING AND TRAILING EDGE COORDINATES
      DO 60 K = 1,NROWS
      YBAR = Y(K)
      IF(YBAR .LT. 0.0) YBAR = -YBAR
      XLEAD(K) = XLB2 * YBAR
      C = CROOT * (1.0-(1.0-TR)*YBAR)
      XTRAIL(K) = XLEAD(K) + C
   60 CONTINUE
C
      RETURN
C*****************************************************************
      END
      SUBROUTINE NORM1
C
C   THIS SUBROUTINE NORMALIZES ALL WING PLANFORM GEOMETRY BY SPAN/2
C
```

```
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
   10 B2 = SPAN / 2.00
      AREA = AREA / B2**2
      CREF = CREF / B2
   20 XMC = XMC / B2
      XCG = XCG / B2
      DO 40 K = 1,NROWS
   30 XLEAD(K) = XLEAD(K) / B2
      XTRAIL(K) = XTRAIL(K) / B2
   40 CONTINUE
      SPAN = 2.00
      ARATIO = SPAN * SPAN / AREA
      RETURN
C***********************************************************************
      END
      SUBROUTINE BOXS(IR)
C
C  THIS SUBROUTINE CONSTRUCTS THE GEOMETRIC PARAMETERS FOR ALL THE
C  EVD ELEMENTS ON THE WING AND JET
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),
     1           NWTYPE,NJTYPE
C
C  CONSTRUCT THE ELEMENTS ON THE WING
C
C  COMPUTE SECTIONAL DATA
   10 CHORD(1) = XTRAIL(1) - XLEAD(1)
      DELTA(1) = 1.00 - Y(1)
      CMAC = CHORD(1)**2 * DELTA(1)
      DO 30 K = 2,NROWS
   20 CHORD(K) = XTRAIL(K) - XLEAD(K)
      DELTA(K) = Y(K-1) - Y(K) - DELTA(K-1)
      IF(DELTA(K) .LT. 0.0) GO TO 190
      CMAC = CMAC + CHORD(K)**2 * DELTA(K)
   30 CONTINUE
C  CHECK THE VALIDITY OF THE SECTIONAL ALLIGNMENT
      YD = Y(NROWS) - DELTA(NROWS)
      IF((ISYMM .GE. 0) .AND. (ABS(YD) .GT. 0.0001)) GO TO 190
      IF((ISYMM .EQ. 1) .AND. (ABS(YD+1.0) .GT. 0.0001)) GO TO 190
      DSUM = DELTA(1)
      DO 35 K = 2,NROWS
      YL = Y(K) + DELTA(K)
      YR = Y(K-1) - DELTA(K-1)
      IF(ABS(YR-YL) .GT. 0.0001) GO TO 190
      DSUM = DSUM + DELTA(K)
   35 CONTINUE
      IF(ABS(DSUM-0.50) .GT. 0.0001) GO TO 190
      CMAC = 2.0 * CMAC / AREA
      IF(ISYMM .LT. 1) CMAC = 2.0 * CMAC
      IF(CREF .LT. 0.0001) CREF = CMAC
      CALL TANS(TANLE,XLEAD,Y,NROWS)
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      I = 0
      DO 90  K = 1,NROWS
C  COMPUTE X-COORDINATES
      NWK = NW(K)
      DO 50 L = 1,NWK
      I = I + 1
      ICK = ICTYPE(K)
   40 XB(I) = XBW(L,ICK)
   50 CONTINUE
C  COMPUTE ALL OTHER PARAMETERS
      I = I - NWK
      IW(K) = I + 1
      DO 80 L = 1,NWK
      I = I + 1
   60 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
   70 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
   80 CONTINUE
C  REDEFINE THE LAST DEL IN THIS SECTION, AND DEFINE THE L.E. EVD TYPE
      DEL(I) = 1.00 - XB(I)
      IWK = IW(K)
      ITYPE(IWK) = 20
   90 CONTINUE
      NWT = I
C
C  CONSTRUCT THE ELEMENTS ON THE JET SHEET
C
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION
      IF(JETFLG .NE. 0) GO TO 180
      DO 170 K = 1,NROWS
C  COMPUTE X-COORDINATES
      IJ(K) = 0
  100 NJK = NJ(K)
      IF(NJK .EQ. 0) GO TO 170
      DO 120 L = 1,NJK
      I = I + 1
      IJK = IJTYPE(K)
  110 XB(I) = XBJ(L,IJK)
```

```
  120 CONTINUE
C  COMPUTE ALL OTHER PARAMETERS
      I = I - NJK
  130 IJ(K) = I + 1
      DO 160 L = 1,NJK
      I = I + 1
  140 KK(I) = K
      DEL(I) = XB(I+1) - XB(I)
  150 XI(I) = XLEAD(K) + XB(I) * CHORD(K)
      ITYPE(I) = 10
  160 CONTINUE
C  REDEFINE THE LAST DEL AND EVD TYPE AND THE D VALUE FOR THIS SECTION
      DEL(I) = 1.0E10
      ITYPE(I) = 30
      D(K) = XI(I) - XTRAIL(K)
  170 CONTINUE
  180 NMAX = I
      IF(NMAX .GT. 600) GO TO 210
      NJT = NMAX - NWT
      IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.
  190 WRITE(6, 200 )
  200 FORMAT(1H1/38X,44HPLEASE CHECK YOUR SECTION LOCATION (Y) INPUT)
      IR = 2
      RETURN

  210 WRITE(6, 220 ) NMAX
  220 FORMAT(1H1/48X,I4,21H IS TOO MANY ELEMENTS)
      IR = 2
      RETURN
      END
      SUBROUTINE BOXJ(NEWMAX,IR)
C
C  THIS SUBROUTINE COMPUTES THE JET BLOWING FACTOR CMUP
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1             D(40),KK(600),ITYPE(600)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
C
C  COMPUTE THE NEW CMUP AND SAVE THE OLD VALUES AS CMUPP
   10 NEWMAX = NMAX
      ICOUNT = 0
      DO 30 K = 1,NROWS
      CMUPP(K) = CMUP(K)
      IF(NJ(K) .EQ. 0) GO TO 30
      IF(CMU(K) .LT. 0.0001) GO TO 20
      CMUP(K) = 2.00 / (CHORD(K)*CMU(K))
      GO TO 30
   20 ICOUNT = ICOUNT + 1
      CMUP(K) = 0.00
   30 CONTINUE
C
      PRINT *, '==> DO YOU WISH TO SEE THE JET BLOWING COEFFICIENTS?'
      PRINT *, '    ENTER (Y OR N)'
   35 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
         WRITE(6, 40 ) (K,CMU(K),K=1,NROWS)
      ELSE IF (ANS.EQ.'N') THEN
         GO TO 45
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 35
      END IF
 1010 FORMAT(A4)
C

   40 FORMAT(1H1,40X,10(4H****)/ 41X,
     1      40H*  SECTIONAL JET BLOWING COEFFICIENTS  */41X,10(4H****)//
     2      53X,3HROW,5X,3HCMU,40(/53X,I2,F12.6))
   45 IF(ICOUNT .EQ. 0) GO TO 50
      IF(ICOUNT .LT. NROWSJ) GO TO 60
      NEWMAX = NWT
   50 IR = 1
      RETURN
C
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND TRY AGAIN.
   60 WRITE(6, 70 )
   70 FORMAT(1H0,43X,35HA ZERO VALUE OF CMU HAS BEEN INPUT.,
     1      33H  THIS CMU CASE HAS BEEN IGNORED.)
      IR = 2
      RETURN
C*********************************************************************
      END
      SUBROUTINE TANS(TAN,X,Y,NROWS)
C
C  THIS SUBROUTINE COMPUTES THE TANGENT OF THE LEADING OR TRAILING EGDE
C  SWEEP ANGLE AT THE CENTERLINE OF EACH SECTION.  IT IS ACCURATE FOR
C  SECTIONS WITH STRAIGHT EDGES IN GROUPS OF THREE OR MORE.
C  IT IS ONLY APPROXIMATE FOR CURVED EDGES.
C  IT MAY RESULT IN ERRORS FOR SECTIONS ADJACENT TO WING BREAKS,
C  IF STRAIGHT EDGES ARE IN ADJACENT GROUPS OF ONLY ONE OR TWO.
C
      DIMENSION TAN(40),X(40),Y(40),S(40)
      SLOP(XR,XL,YR,YL) = (XR-XL) / (YR-YL)
C
      DO 50 K = 1,NROWS
      KR = K-1
      KL = K
```

```
      IF(K .GT. 1) GO TO 30
      KR = 1
      KL = 2
   30 S(K) = SLOP(X(KR),X(KL),Y(KR),Y(KL))
   50 CONTINUE
      DO 200 K = 1,NROWS
      IF(K .LT. 3) GO TO 150
      IF(K .EQ. NROWS) GO TO 150
      IF(K .EQ. (NROWS-1)) GO TO 160
C  CHECK WHETHER THE RIGHT OR LEFT SIDES ARE STRAIGHT
      IF(ABS(S(K) - S(K-1)) .LT. 0.001) GO TO 150
      IF(ABS(S(K+1) - S(K+2)) .LT. 0.001) GO TO 160
C  NEITHER SIDE IS CONCLUSIVELY STRAIGHT - CHECK FURTHER LEFT AND RIGHT
      IF(K .EQ. 3) GO TO 160
      IF(K .EQ. (NROWS-2)) GO TO 150
      IF(ABS(S(K-1) - S(K-2)) .LT. 0.001) GO TO 160
      IF(ABS(S(K+2) - S(K+3)) .LT. 0.001) GO TO 150
C  THE TRUE SHAPE CANNOT BE DETERMINED - GIVE UP AND TAKE THE AVERAGE
      TAN(K) = (S(K) + S(K+1)) / 2.00
      GO TO 200
C  THE RIGHT EDGE IS STRAIGHT
  150 TAN(K) = S(K)
      GO TO 200
C  THE LEFT EDGE IS STRAIGHT
  160 TAN(K) = S(K+1)
  200 CONTINUE
      RETURN
C*****************************************************************
      END
      SUBROUTINE INCASE(LCASE,NOALFA)
C
C  THIS SUBROUTINE READS THE FUNDAMENTAL GEOMETRIC CASE DATA
C
      CHARACTER*1 ANS
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/FCASE1/INTWST,INHITE,INDELJ,INCAMB,INBETA
      COMMON/FCASE2/TWIST(40,10),HL(40,10),DJ(40,10),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/INDAT/LUN
      DIMENSION NI(10),DUMMY(40)
C
      IF((LCASE .EQ. 1) .AND. (NOALFA .GT. 0)) RETURN
C
      CALL CLRSCRN
      PRINT *
      WRITE(6,5) LCASE
    5 FORMAT(1X,4X,'FUNDAMENTAL CASE CONTROL FLAGS FOR CASE ',I2,'.')
      PRINT *
      PRINT *, '==> THE FOLLOWING QUESTIONS ARE USED TO SET THE CONTROL
     + FLAGS.'
      PRINT *, '    THESE FLAGS IDENTIFY THE TYPES OF LINEAR GEOMETRIC V
     +ARIATIONS'
      PRINT *, '    TO BE INCLUDED IN EACH FUNDAMENTAL CASE.'
      PRINT *, '    THE ANGLE OF ATTACK CASE IS ALREADY INCLUDED AS CASE
     + 1.'
      PRINT *
      PRINT *, '    A NO RESPONSE INDICATES THAT THE VARIATION WILL BE O
     +MITTED.'
      PRINT *
      PRINT *, '    A YES RESPONSE INDICATES THAT THE VARIATION WILL BE
     +INCLUDED'
      PRINT *, '    AND THAT YOU WILL PROVIDE THE REQUIRED AMPLIFYING IN
     +FORMATION.'
      PRINT *
   10 CONTINUE
C-----------------------------------------------------------------
C  READ FUNDAMENTAL CASE CONTROL FLAGS
C-----------------------------------------------------------------
C
      PRINT *, '==> VARY SPANWISE TWIST DISTRIBUTION? (Y OR N)'
   20 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
         INTWST = LCASE
      ELSE IF (ANS.EQ.'N') THEN
         INTWST = 0
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 20
      END IF
      PRINT *
      PRINT *, '==> VARY LEADING EDGE VERTICAL DISPLACEMENT? (Y OR N)'
   30 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
         INHITE = LCASE
      ELSE IF (ANS.EQ.'N') THEN
         INHITE = 0
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 30
      END IF
      PRINT *
      PRINT *, '==> VARY JET DEFLECTION? (Y OR N)'
   40 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
         INDELJ = LCASE
      ELSE IF (ANS.EQ.'N') THEN
         INDELJ = 0
      ELSE
         PRINT *, '  INVALID RESPONSE - REENTER.'
         GO TO 40
      END IF
```

```
      PRINT *
      PRINT *, '==> VARY THE WING CAMBER? (Y OR N)'
   50 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
        INCAMB = LCASE
      ELSE IF (ANS.EQ.'N') THEN
        INCAMB = 0
      ELSE
        PRINT *, '  INVALID RESPONSE - REENTER.'
        GO TO 50
      END IF
      PRINT *
      PRINT *, '==> VARY THE WING HINGE DEFLECTION? (Y OR N)'
   60 READ (5,'(A1)') ANS
      IF (ANS.EQ.'Y') THEN
        INBETA = LCASE
      ELSE IF (ANS.EQ.'N') THEN
        INBETA = 0
      ELSE
        PRINT *, '  INVALID RESPONSE - REENTER.'
        GO TO 60
      END IF
      PRINT *
C-----------------------------------------------------------------------
C  SUMMARY OF FUNDAMENTAL CASE CONTROL FLAGS DATA
C-----------------------------------------------------------------------
      CALL CLRSCRN
      WRITE (6,580)
580   FORMAT (1X,'SUMMARY OF FUNDAMENTAL CASE CONTROL FLAGS DATA?',
     1/,1X,25H==> ENTER Y = YES; N = NO)
      READ (5,'(A1)') ANS
      IF (ANS.EQ.'N') GO TO 70
         CALL CLRSCRN
         WRITE(6,6) LCASE
    6    FORMAT(1X,2X,'CONTROL FLAGS FOR FUNDAMENTAL CASE ',I2'.')
         PRINT *
         PRINT *, '  A NONZERO FLAG INDICATES THAT THE LINEAR VARIATION'
         PRINT *, '  WILL BE INCLUDED.  THE VALUE OF A NONZERO FLAG'
         PRINT *, '  HAS BEEN SET TO THE FUNDAMENTAL CASE IN WHICH IT'
         PRINT *, '  IS INCORPORATED, HOWEVER THIS CHOICE IS ARBITRARY.'
         PRINT *
         WRITE (6,581)
         WRITE (6,582) INTWST,INHITE,INDELJ,INCAMB,INBETA
         WRITE (6,590)
         READ (5,'(A1)') ANS
         IF (ANS.EQ.'Y') GO TO 10
581   FORMAT (1X,'INTWST',5X,'INHITE',5X,'INDELJ',5X,'INCAMB',5X,
     +'INBETA')
582   FORMAT (1X,5(2X,I2,7X))
590   FORMAT (//,1X,43HCHANGE FUNDAMENTAL CASE CONTROL FLAGS DATA?,
     1/,1X,25H==> ENTER Y = YES; N = NO)
 70   CONTINUE
C
C  WRITE TO DATA FILE
      WRITE(LUN, 601 ) INTWST,INHITE,INDELJ,INCAMB,INBETA
  601 FORMAT(5I2)
C-----------------------------------------------------------------------
C  READ SECTIONAL TWIST, HEIGHT AND JET DEFLECTION DATA
      IF(INTWST .EQ. 0) GO TO 85
C-----------------------------------------------------------------------
C  TWIST DISTRIBUTION CARDS
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *,'     SPANWISE WING TWIST DISTRIBUTION VALUES'
      PRINT *
      PRINT *,'     THE SECTIONAL TWIST IS THE WING TWIST AT THE SECTION'
      PRINT *,'     CENTERLINE WITH RESPECT TO THE WING REFERENCE PLANE.'
      PRINT *
      PRINT *,'     POSITIVE VALUES ARE IN THE SAME SENSE AS A POSITIVE'
      PRINT *,'     ANGLE-OF-ATTACK (LEADING EDGE UP).'
      PRINT *
      PRINT *,'==> ENTER TWIST, SECTIONAL WING TWIST, IN DEGREES.(R)'
      PRINT *
      DO 80 K = 1,NROWS
      WRITE(6,12) K,NROWS
      READ(5, * ) TWIST(K,LCASE)
   12 FORMAT(1X,'  ENTER SECTION TWIST FOR SECTION ',I2,' OF ',I2,' SECT
     +IONS ',/)
   80 CONTINUE
      PRINT *
C SUMMARY REQD
C
C WRITE TO DATA FILE
      WRITE(LUN, 701 ) (TWIST(K,LCASE),K=1,NROWS)
  701 FORMAT(8F10.6)
C
   85 IF(INHITE .EQ. 0) GO TO 95
C-----------------------------------------------------------------------
C  LEADING EDGE VERTICAL DISPLACEMENT CARDS
C-----------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *,'     LEADING EDGE VERTICAL DISPLACEMENT'
      PRINT *
      PRINT *,'     THIS DATA INDICATES THE VERTICAL DISPLACEMENT OF THE'
      PRINT *,'     LEADING EDGE FROM THE WING REFERENCE PLANE. VALUES'
      PRINT *,'     MUST BE NORMALIZED BY THE SECTIONAL CHORD.'
      PRINT *
      PRINT *,'     DISPLACEMENT MAY BE THE RESULT OF DIHEDRAL, TWIST,'
      PRINT *,'     NONLINEAR MOVEMENT OF A LEADING EDGE DEVICE, ETC.'
```

```
      PRINT *,'    TRANSLATION DUE TO ORDINARY LINEAR LEADING AND'
      PRINT *,'    TRAILING FLAP DEFLECTIONS AND ANGLE OF ATTACK ARE'
      PRINT *,'    ACCOUNTED FOR AUTOMATICALLY BY THE PROGRAM.'
      PRINT *
      PRINT *,'==> ENTER HL, NORMALIZED LEADING EDGE DISPLACEMENT.(R)'
      PRINT *
      DO 90 K = 1,NROWS
      WRITE(6,22) K,NROWS
      READ(5, * ) HL(K,LCASE)
   22 FORMAT(1X,'  ENTER DISPLACEMENT FOR SECTION ',I2,' OF ',I2,
     + ' SECTIONS.',/)
   90 CONTINUE
      PRINT *
C
C SUMMARY REQD
C
C WRITE TO DATA FILE
      WRITE(LUN, 701 ) (HL(K,LCASE),K=1,NROWS)
C
   95 IF(INDELJ .EQ. 0) GO TO 105
C------------------------------------------------------------------
C  JET DEFLECTION CARDS
C------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *,'    JET DEFLECTION'
      PRINT *
      PRINT *,'    THIS DATA INDICATES THE SPANWISE VARIATION OF JET'
      PRINT *,'    DEFLECTION RELATIVE TO THE TRAILING EDGE.  THE JET'
      PRINT *,'    TURNING ANGLE IS MEASURED RELATIVE TO THE MEAN LINE'
      PRINT *,'    OF THE TRAILING EDGE.  VALUES ARE INPUT WORKING'
      PRINT *,'    FROM THE RIGHTMOST JET TOWARDS THE CENTERLINE.'
      PRINT *
      PRINT *,'    A DOWNWARD DEFLECTION IS DEFINED AS POSITIVE.'
      PRINT *
      PRINT *,'==> ENTER DJ, THE JET TURNING ANGLE, IN DEGREES.(R)'
      PRINT *
      DO 100 K = 1,NROWSJ
      WRITE(6,32) K,NROWSJ
      READ(5, * ) DJ(K,LCASE)
   32 FORMAT(1X,'  ENTER DEFLECTION FOR JET SECTION ',I2,' OF ',I2,
     + ' JET SECTIONS.',/)
  100 CONTINUE
      PRINT *
C
C SUMMARY REQD
C
C WRITE TO DATA FILE
      WRITE(LUN, 701 ) (DJ(K,LCASE),K=1,NROWSJ)
C
  105 IF(INCAMB .EQ. 0) GO TO 160
C  INPUT CAMBER TYPE OF EACH SECTION
C------------------------------------------------------------------
C  WING SECTION CAMBER TYPE CARDS
C------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '    WING SECTION CAMBER TYPES'
      PRINT *
      PRINT *, '    THIS DATA IS SIMILAR TO THE WING SECTION TYPE DATA.'
      PRINT *, '    IN ORDER FOR SECTIONS TO BE OF THE SAME CAMBER TYPE'
      PRINT *, '    THEY MUST BE OF THE SAME WING SECTION TYPE (ICTYPE)'
      PRINT *, '    AND THE CAMBER ANGLES ASSOCIATED WITH EACH ELEMENT'
      PRINT *, '    MUST BE THE SAME. BEGIN WITH A TYPE NUMBER OF 1 AND'
      PRINT *, '    WORK IN SEQUENCE, 2,3,...(ASCENDING ORDER).(I)'
      PRINT *
      PRINT *, '    A SECTION WITH NO CAMBER HAS A TYPE OF 0 (ZERO).'
      PRINT *
      PRINT *, '    A MAXIMUM OF 10 CAMBER TYPES IS ALLOWED.'
      PRINT *
      PRINT *, '==> ENTER ICT, THE CAMBER TYPE NUMBER OF EACH SECTION.'
      PRINT *
      NCT = 0
      DO 110 K = 1,NROWS
      WRITE(6,42) K,NROWS
      READ(5, * ) ICT(K)
C  DETERMINE NUMBER OF CAMBER TYPES
      IF(ICT(K) .EQ. 0) GO TO 110
      IF(ICT(K) .GT. NCT) NCT = ICT(K)
      ICK = ICT(K)
      NI(ICK) = NW(K)
      IF(NCT .GT. 8) THEN
         WRITE(6,41) NCT
         PRINT *, '    A MAXIMUM OF 10 CAMBER TYPES IS ALLOWED.'
         PRINT *, '          ***** W A R N I N G *****'
         PRINT *, '    YOU MAY ENTER ONLY ONE MORE DIFFERENT TYPE.'
      END IF
   41 FORMAT(1X,5X,29HNUMBER OF WING CAMBER TYPES =,I3)
   42 FORMAT(1X,'  ENTER CAMBER TYPE FOR SECTION ',I2,' OF ',I2,
     +' SECTIONS.',/)
  110 CONTINUE
C
C  SUMMARY REQD
C
C WRITE TO DATA FILE
      WRITE(LUN, 101 ) (ICT(K),K=1,NROWS)
  101 FORMAT(40I2)
C------------------------------------------------------------------
C  CAMBER ANGLES FOR EACH CAMBER SECTION TYPE
C------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
```

```
      PRINT *, '     CAMBER ANGLES FOR THE DOWNWASH CONTROL POINTS'
      PRINT *
      PRINT *, '     THE CAMBER ANGLE FOR THE DOWNWASH CONTROL POINT OF'
      PRINT *, '     EACH EVD ELEMENT IS REQUIRED.  THE DOWNWASH CONTROL'
      PRINT *, '     POINT IS ARBITRARILY CHOSEN AS HALFWAY BETWEEN ANY'
      PRINT *, '     TWO ADJACENT XB(EVD BOUNDARY) POINTS, INCLUDING THE'
      PRINT *, '     TRAILING EDGE.'
      PRINT *
      PRINT *, '     POSITIVE VALUES ARE IN THE SAME SENSE AS A POSITIVE'
      PRINT *, '     ANGLE-OF-ATTACK (LEADING EDGE UP).'
      PRINT *
      PRINT *, '==> ENTER AC, THE CAMBER ANGLE, IN DEGREES.(R)'
      PRINT *
C  READ THE CHORDWISE CAMBER ANGLES FOR EACH CAMBER TYPE
      DO 130 N = 1,NCT
      NIN = NI(N)
          DO 125 L = 1,NIN
            WRITE(6,52) NCT
            WRITE(6,53) L,NIN
            READ(5, * ) AC(L,N)
  125     CONTINUE
  130 CONTINUE
   52 FORMAT(1X,'  FOR CAMBER SECTION TYPE NUMBER ',I2)
   53 FORMAT(1X,'  ENTER CAMBER ANGLE FOR EVD ELEMENT ',I2,' OF ',I2,/)
C   54 FORMAT(1X,'  LOCATED AT CHORDWISE COORDINATE =',F10.6)
C
C  SUMMARY REQD
C
C WRITE TO DATA FILE
      DO 135 N = 1,NCT
      NIN = NI(N)
      WRITE(LUN, 701 ) (AC(L,N),L=1,NIN)
  135 CONTINUE
      IF(NROWSJ .EQ. 0) GO TO 160
C---------------------------------------------------------------------
C  TRAILING EDGE CAMBER ANGLE DATA CASE WITH JETS AND CAMBER
C---------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '     TRAILING EDGE CAMBER ANGLE FOR WINGS WITH'
      PRINT *, '     JET SHEETS AND CAMBER.'
      PRINT *
      PRINT *, '     THE TRAILING EDGE DEFLECTION ANGLE DUE TO CAMBER'
      PRINT *, '     ONLY IS ENTERED HERE.  THESE VALUES ARE USED TO'
      PRINT *, '     DETERMINE THE TOTAL JET DEFLECTION ANGLE WITH'
      PRINT *,'     RESPECT TO THE FREESTREAM.  VALUES ARE INPUT WORKING'
      PRINT *,'     FROM THE RIGHTMOST JET TOWARDS THE CENTERLINE.'
      PRINT *
      PRINT *, '     POSITIVE VALUES ARE IN THE SAME SENSE AS A POSITIVE'
      PRINT *, '     ANGLE-OF-ATTACK (LEADING EDGE UP).'
      PRINT *
      PRINT *,'==> ENTER ACTE, TRAILING EDGE CAMBER ANGLE,(DEGREES).(R)'
      PRINT *
C  READ THE TRAILING EDGE CAMBER ANGLE FOR EACH JET SECTION
      DO 140 K = 1,NROWSJ
         WRITE(6,62) K
         WRITE(6,63)
         READ(5, * ) ACTE(K)
  140 CONTINUE
   62 FORMAT(1X,'  FOR JET SECTION NUMBER ',I2)
   63 FORMAT(1X,'  ENTER CAMBER ANGLE FOR TRAILING EDGE ',/)
C
C SUMMARY REQD
C
C WRITE TO DATA FILE
      WRITE(LUN, 701 ) (ACTE(K),K=1,NROWSJ)
C
C  STOPPED HERE (JAC) - CASES WITH JETS HAVE NOT BEEN FINISHED.
C---------------------------------------------------------------------
C  THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA
C---------------------------------------------------------------------
C  READ THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA
  160 IF(INBETA .EQ. 0) GO TO 210
C  170 READ(5, 100 ) (IHT(K),K=1,NROWS)
      NHT = 0
      DO 180 K = 1,NROWS
      IF(IHT(K) .GT. NHT) NHT = IHT(K)
  180 CONTINUE
      DO 200 N = 1,NHT
      READ(5, 190 ) (XHB(L,N),IFS(L,N),BET(L,N),L=1,4)
  190 FORMAT(4(F10.6,I1,F9.6))
  200 CONTINUE
C  170 WRITE(5, 100 ) (IHT(K),K=1,NROWS)
  210 RETURN
C*********************************************************************
      END
      SUBROUTINE OUT1(LCASE)
C
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE
C  SECTIONAL METHOD INPUT
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /LUKE/ TITLE(20)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(600),XI(600),DEL(600),
     1            D(40),KK(600),ITYPE(600)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TWIST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1            XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)
```

```
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI
C
C  PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS
      IF(LCASE .GT. 1) GO TO 60
   10 WRITE(6, 20 )TITLE
   20 FORMAT(1H1,39X,10(4H****)/
     1    40X,40H*  EVD  JET - WING  COMPUTER  PROGRAM  */
     2    40X,10(4H****)///20X,20A4)
      CMA = CMAC * SPA / 2.0
   30 WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,
     1             ARATIO,XCG,XC
   40 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /
     1          41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /
     2          41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /
     3          41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /
     4          42X,5HXCG =,2F15.6)
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,
     1             IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX
   50 FORMAT(1H0/ 48X,7HNROWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /
     1            48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /
     2            47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /
     3            47X,8HIHINGE =,I3,7X,I3 ///
     4            43X,25HNUMBER OF WING ELEMENTS =,I4 /
     5            43X,25HNUMBER OF JET ELEMENTS  =,I4 /
     6            42X,26HTOTAL NUMBER OF ELEMENTS =,I4)
   60 J = 0
      JJ = NWT
C
C  PRINT FUNDAMENTAL CASE HEADER
      WRITE(6, 70 ) LCASE
   70 FORMAT(1H1,23X,1H*,19(4H****)/
     1     24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,
     2     17H FUNDAMENTAL CASE,I3,3H  */24X,1H*,19(4H****))
      ILINES = 3
      DO 260 K = 1,NROWS
C
C  PRINT SECTIONAL DATA
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)
   80 FORMAT(1H0,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,
     2 2X,7HTANLE =,F10.6)
C
C  PRINT CHORDWISE DATA ON WING
      NWK = NW(K)
      WRITE(6, 90 ) NWK,TWIST(K,LCASE),HL(K,LCASE),THS(K,LCASE)
   90 FORMAT(1H0,21HWING ELEMENTS    NW =,I3,5X, 7HTWIST =,F10.6,5X,
     1             4HHL =,F10.6,5X, 9HTHETA S =,F10.6)
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)
  100 FORMAT(1H ,14X,2HXB,10F11.6 / 17X,10F11.6)
      IF(LCASE .GT. 1) GO TO 130
      WRITE(6, 110 ) (XI(J+L),L=1,NWK)
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)
  130 IF(ICT(K) .EQ. 0) GO TO 150
      ICK = ICT(K)
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)
  140 FORMAT(1H ,10X,6HCAMBER,10F11.6/17X,10F11.6)
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))
      J = J + NWK
      IL = 1
      IF(NWK .GT. 9) IL = 2
      ILINES = ILINES + 4 + 4*IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
C
C  PRINT CHORDWISE DATA ON JET
      NJK = NJ(K)
      IF(NJK .GT. 0) GO TO 200
      WRITE(6, 190 )
  190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)
      ILINES = ILINES + 1
      GO TO 230
  200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)
  210 FORMAT(1H0,1X,20HJET ELEMENTS     NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,
     1     F10.6,5X,6HACTE =,F10.6,5X,7HTHETA =,F10.6)
      WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)
      IF(LCASE .GT. 1) GO TO 220
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)
      JJ = JJ + NJK
      IL = 1
      IF(NJK .EQ. 10) IL = 2
      ILINES = ILINES + 1 + 3 * IL
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL
  230 IF(K .EQ. NROWS) GO TO 260
      NWK1 = NW(K+1)
      IL = 1
      IF(NWK1 .GT. 9) IL = 2
      NEXT = 4 + 4*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
      NJK1 = NJ(K+1)
      IL = 1
      IF(NJK1 .EQ. 10) IL = 2
      NEXT = NEXT + 1
```

```
      IF(NJK1 .EQ. 0) GO TO 240
      NEXT = NEXT + 1 + 3*IL
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL
  240 IF((55-ILINES) .GE. NEXT) GO TO 260
      WRITE(6, 250 )
  250 FORMAT(1H1)
      ILINES = 1
  260 CONTINUE
      RETURN
C*******************************************************************
      END
      SUBROUTINE INCOMP(NCASES,IR)
C
C  THIS SUBROUTINE READS IN THE COMPOSITE CASE REQUIREMENTS
C  WHICH DEFINE THE FUNDAMENTAL CASES AND THEIR DEFLECTION MAGNITUDE
C  FOR SUPERPOSITION IN UP TO 24 COMBINATIONS
C
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/INDAT/LUN
      DIMENSION FUNNY(10),ND(10),NFC(10)
C
C-------------------------------------------------------------------
C  COMPOSITE CASE REQUIREMENTS CARDS
C-------------------------------------------------------------------
      CALL CLRSCRN
      PRINT *
      PRINT *, '      COMPOSITE CASES'
      PRINT *
      PRINT *, '      THE FOLLOWING INFORMATION SPECIFIES HOW THE DATA'
      PRINT *, '      FOR FOR THE FUNDAMENTAL CASES INPUT ON THE PREVIOUS'
      PRINT *, '      CARDS, IS TO BE COMBINED TO FORM OR MODEL THE WING'
      PRINT *, '      UNDER STUDY.  A MAXIMUM OF 24 COMPOSITE CASES MAY'
      PRINT *, '      BE REQUESTED.'
      PRINT *, '      YOU WILL BE ASKED FOR FUNDAMENTAL CASE NUMBERS AND'
      PRINT *, '      THE MULTIPLICATIVE FACTOR TO BE APPLIED TO EACH.'
      PRINT *
      PRINT *, '      THE FUNDAMENTAL CASES ARE IDENTIFIED IN THE SAME'
      PRINT *, '      SEQUENCE AS THEY WERE INPUT, 1,2,3...'
      PRINT *
      PRINT *, '      IF A MULTIPLICATIVE FACTOR OF 1.5 IS APPLIED TO A'
      PRINT *, '      A FUNDAMENTAL CASE WITH A HINGE DEFLECTION OF 10'
      PRINT *, '      DEGREES, THE COMPOSITE CASE WILL HAVE 15 DEGREES.'
      PRINT *
C
C  READ THE COMPOSITE CASE DATA, CONSISTING OF FUNDAMENTAL CASE
C  DEFLECTIONS, IN DEGREES
      NCC = 0
C ENTER THE NUMBER OF COMPOSITE CASES TO BE INPUT
      PRINT *, '      HOW MANY COMPOSITE CASES WILL YOU BE ENTERING?(I)'
      PRINT *, '      (FOR NO COMPOSITE CASES ENTER ZERO.)'
   10 READ(5, * ) NCC
      IF (NCC .EQ. 0) GO TO 100
      IF (NCC .GT. 24) THEN
        WRITE(6,110)
        PRINT *, '          PLEASE REENTER'
        GO TO 10
      END IF
      DO 15 I = 1,NCC
C ENTER THE NUMBER OF FUNDAMENTAL CASES TO BE COMBINED ON THIS CASE
      PRINT *, '      HOW MANY FUNDAMENTAL CASES WILL BE COMBINED'
      PRINT *, '      TO MAKE UP THIS COMPOSITE CASE? (MAXIMUM OF 10)'
   20 READ(5, * ) NFC(I)
      NFCI = NFC(I)
      IF (NFCI .GT. 10) THEN
        WRITE(6,120)
        PRINT *, '          PLEASE REENTER'
        GO TO 20
      END IF
C  READ IN THE DATA FOR EACH CARD
      DO 30 N = 1,NFCI
      WRITE(6,32) I
      WRITE(6,33) N,NFCI
   35 READ(5, * ) ND(N)
C  CHECK THE VALIDITY OF THE DATA
      IF(ND(N) .GT. NCASES) THEN
        WRITE(6,71) NCASES
        PRINT *, '          PLEASE REENTER'
        GO TO 35
      END IF
      WRITE(6,44)
      READ(5, * ) FUNNY(N)
      PRINT *
   30 CONTINUE
   15 CONTINUE
   32 FORMAT(1X,'        FOR COMPOSITE CASE ',I2)
   33 FORMAT(1X,'   ENTER FUNDAMENTAL CASE ',I2,' OF ',I2,'.',/)
   44 FORMAT(1X,'   ENTER THE MULTIPLICATIVE FACTOR FOR THIS CASE.',I2,/)
   71 FORMAT(1X,5X,'THE FUNDAMENTAL CASE VALUE CANNOT BE GREATER THAN ',
     1I2,/)
C  TOO MANY COMPOSITE CASES HAVE BEEN REQUESTED.  REENTER.
  110 FORMAT(1X,5X,'A MAXIMUM OF 24 COMPOSITE CASES MAY BE INPUT.')
  120 FORMAT(1X,5X,'A MAXIMUM OF 10 FUNDAMENTAL CASES MAY BE INCLUDED'
     1/,1X,5X,'ON ANY ONE COMPOSITE CASE.')
C
C  SUMMARY GOES HERE
C
C
C
C  WRITE DATA TO FILE
      DO 70 I = 1,NCC
      NFCI = NFC(I)
      WRITE(LUN, 40 ) (ND(L),FUNNY(L),L=1,NFCI)
```

```
   40 FORMAT(10(BZ,I2,F6.4))
   70 CONTINUE
C
C  OUTPUT A 9 CARD AFTER NCC SETS OF DATA HAVE BEEN INPUT
  100 WRITE(LUN, 102 )
  102 FORMAT('9    ')
C
      IR = 1
      RETURN
C*********************************************************************
      END
      SUBROUTINE BLOWIN(JETFLG,IR)
C
C  THIS SUBROUTINE READS THE SECTIONAL JET BLOWING RATES
C  CMU(K) = J / (Q * CHORD(K))
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      DIMENSION DCMU(40)
      COMMON/INDAT/LUN
C
      IF(JETFLG .NE. 0) GO TO 30
C  READ THE CMU DATA ONLY FOR THOSE SECTIONS WHICH HAVE A JET
      READ(5, 10, END=60 ) (DCMU(K),K=1,NROWSJ)
   10 FORMAT(8F10.6)
   20 IF(DCMU(1) .LT. 800.0) GO TO 30
      IR = 2
      RETURN
C
C  REARRANGE THE DATA INTO THE PROPER SEQUENCE
   30 KP = 0
      DO 50 K = 1,NROWS
   40 CMU(K) = 0.00
      IF(NJ(K) .EQ. 0) GO TO 50
      KP = KP + 1
      CMU(K) = DCMU(KP)
   50 CONTINUE
      IR = 1
      RETURN
C
C  AN END OF FILE HAS BEEN READ.  THIS RUN IS COMPLETELY FINISHED.
   60 WRITE(6, 70 )
   70 FORMAT(1H1///41X,37HNO MORE CMU CASES HAVE BEEN REQUESTED)
      IR = 3
      RETURN
      END
C
C*********************************************************************
```

# APPENDIX E. FIGURES GENERATED USING DISSPLA

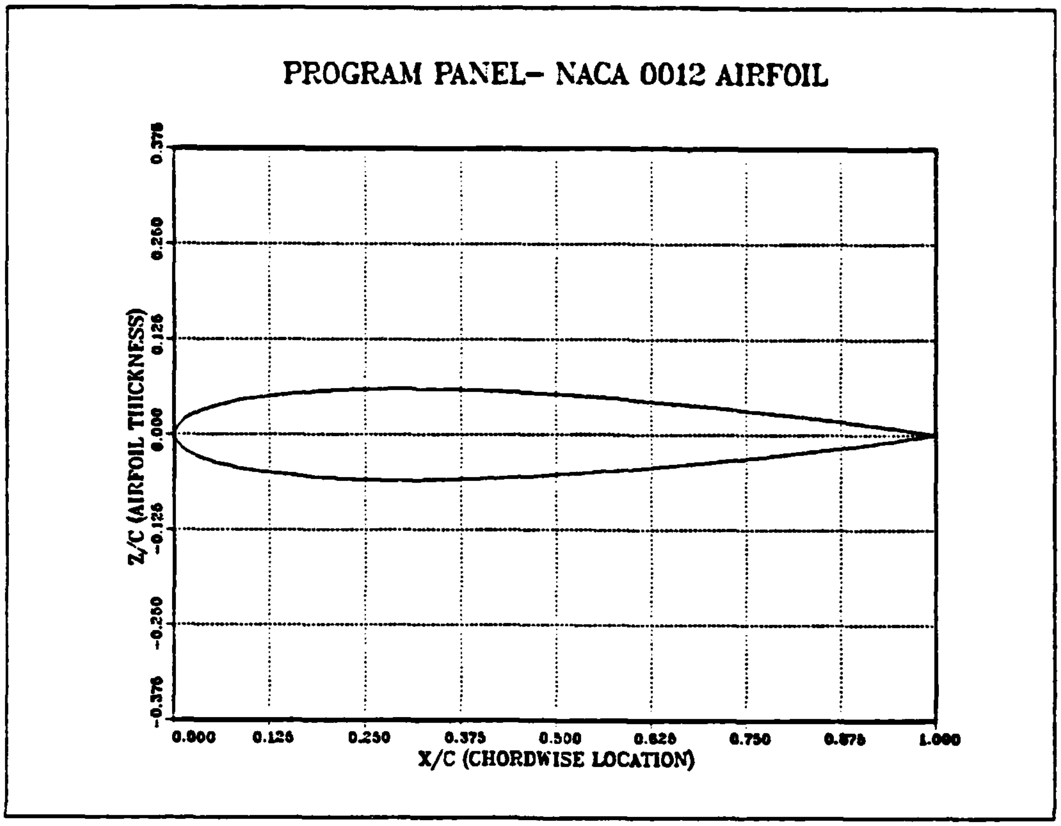


**Figure 26. Program PANEL- Shape Generated Using Airfoil Coordinates Data File**

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The surface coordinates for the airfoil were input to the PANEL program using an input data file containing 28 surface points.

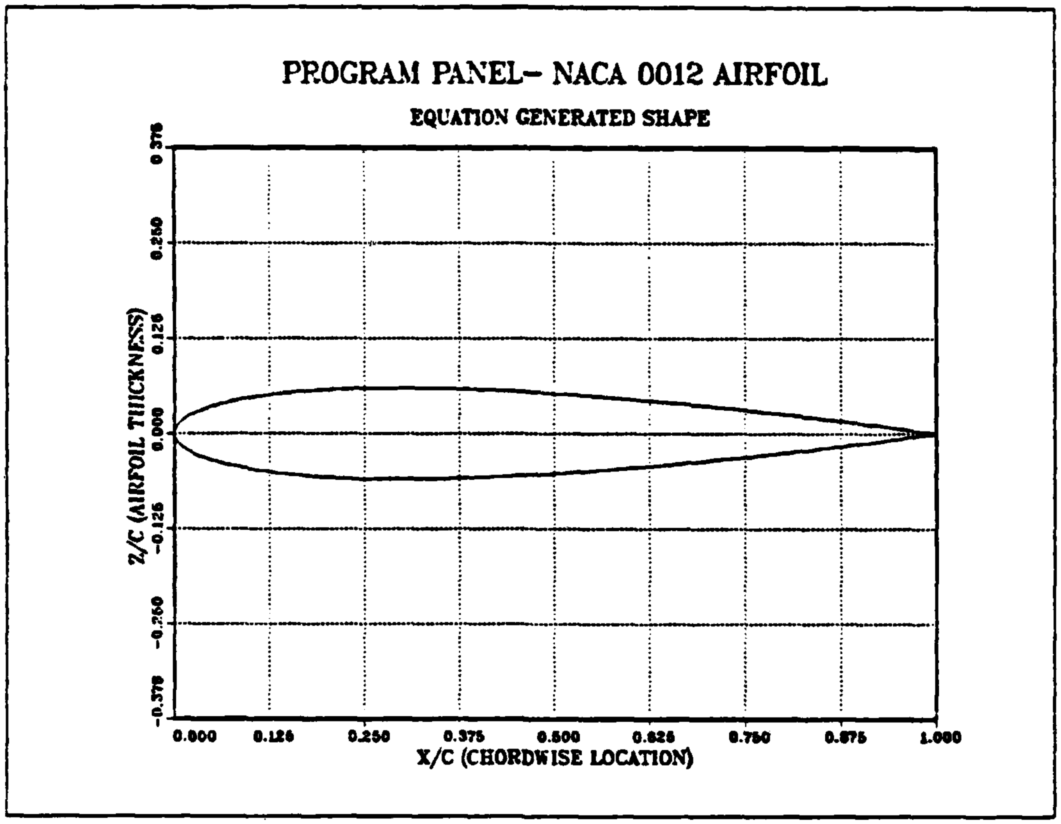


**Figure 27. Program PANEL- Shape Generated Using Internal Equation for NACA 0012**

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The surface coordinates for the airfoil were generated by the PANEL program using the internal equation for NACA XXXX series airfoils. Twenty points were used to describe the surface. Despite using fewer points to define the surface, there is virtually no difference between this plot and the one on the preceding page which used actual airfoil surface data.

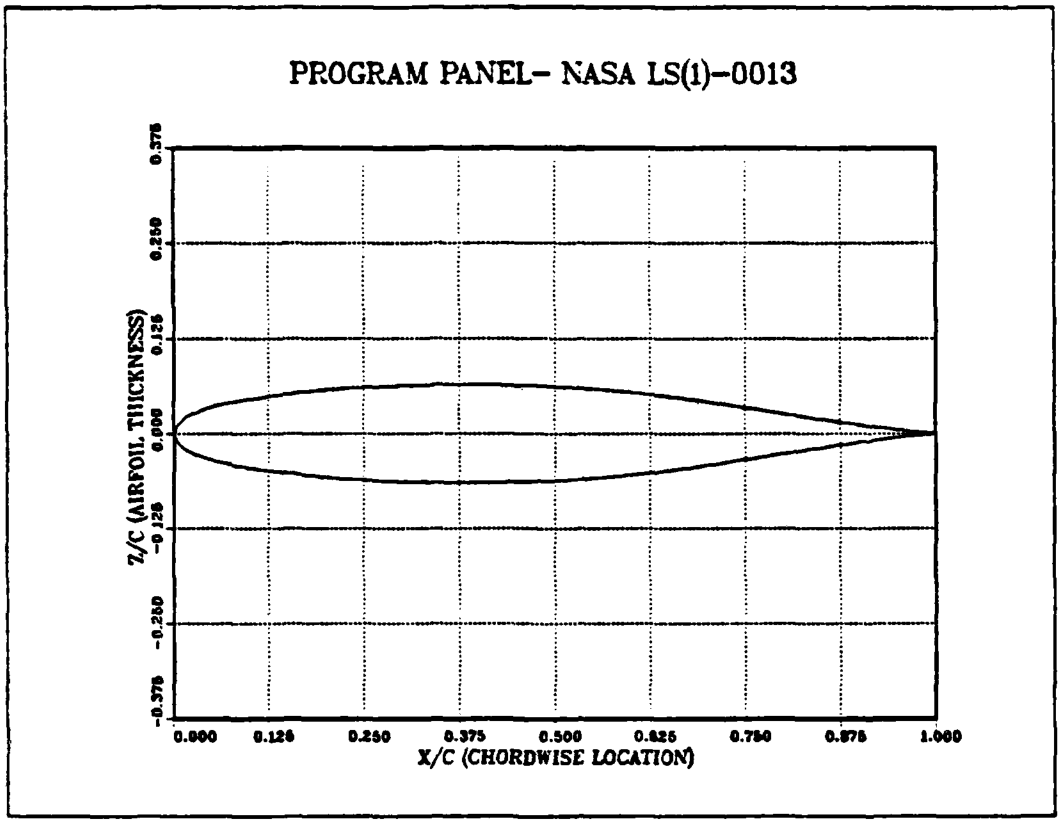


**Figure 28. Program PANEL- Shape Generated Using DATA Statements for NASA LS(1)-0013**

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The surface coordinates for the airfoil were input to the PANEL program using the DATA statement entry method. The DATA statements for the NASA LS(1)-0013 within the PANEL program contain coordinates for 28 surface locations. This plot is nearly identical to that found in Ref. 18.

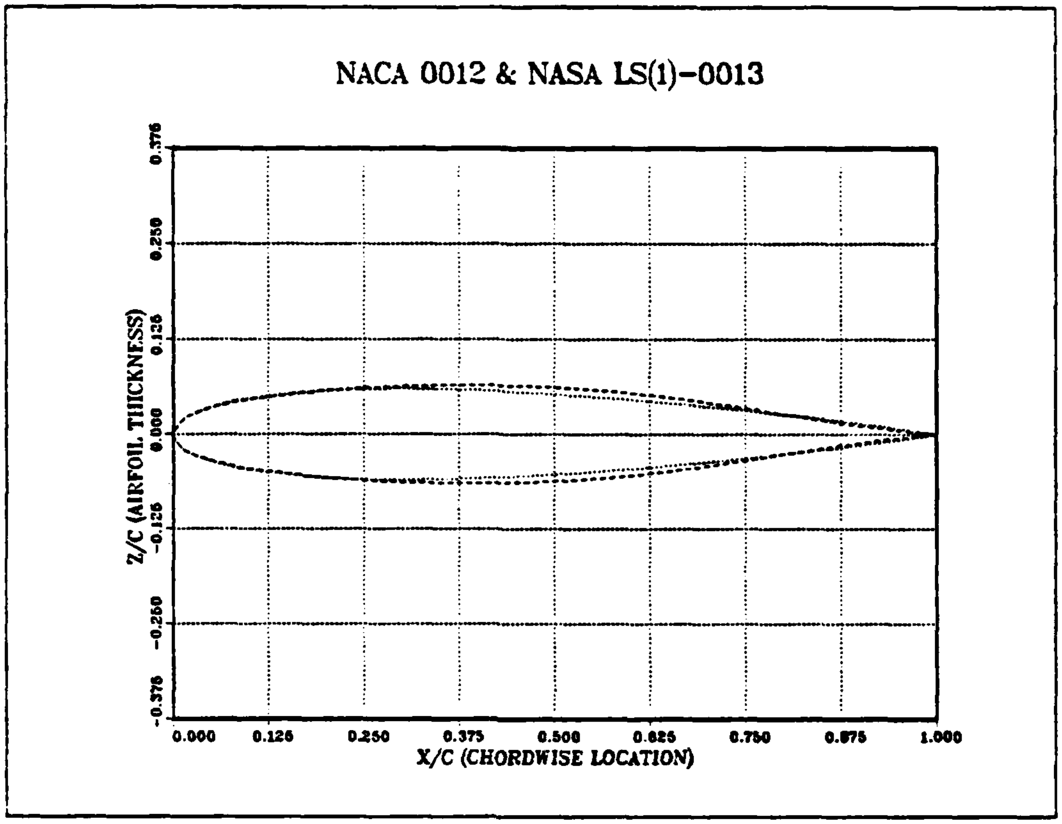


**Figure 29. Program PANEL- Comparison of Shapes Generated for NACA 0012 and NASA LS(1)-0013**

This figure compares the shapes of the NACA 0012 and NASA LS(1)-0013 airfoils. The actual surface coordinates were used for this plot. Again, this plot is nearly identical to a similar plot found in Ref. 18.

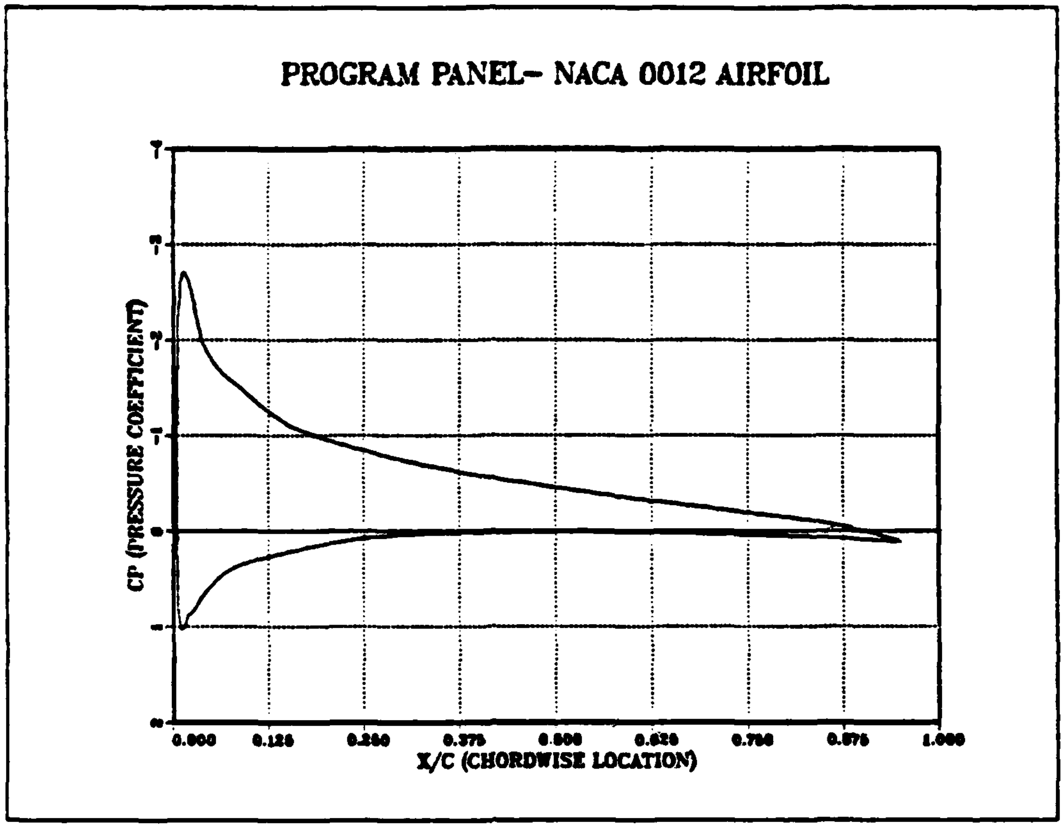


Figure 30. Program PANEL-Surface Pressure Distribution for NACA 0012

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The pressure distribution is for the NACA 0012 airfoil defined by an input data file containing 28 surface points at an angle of attack of six degrees. The results of the program run are repeated below.

```
DATA FILE:  PPRESS.DAT
ANGLE OF ATTACK IN DEGREES =     6.000

   CD = 0.00387    CL = 0.70980    CM =-0.17750    CMC4 =-0.00092
```

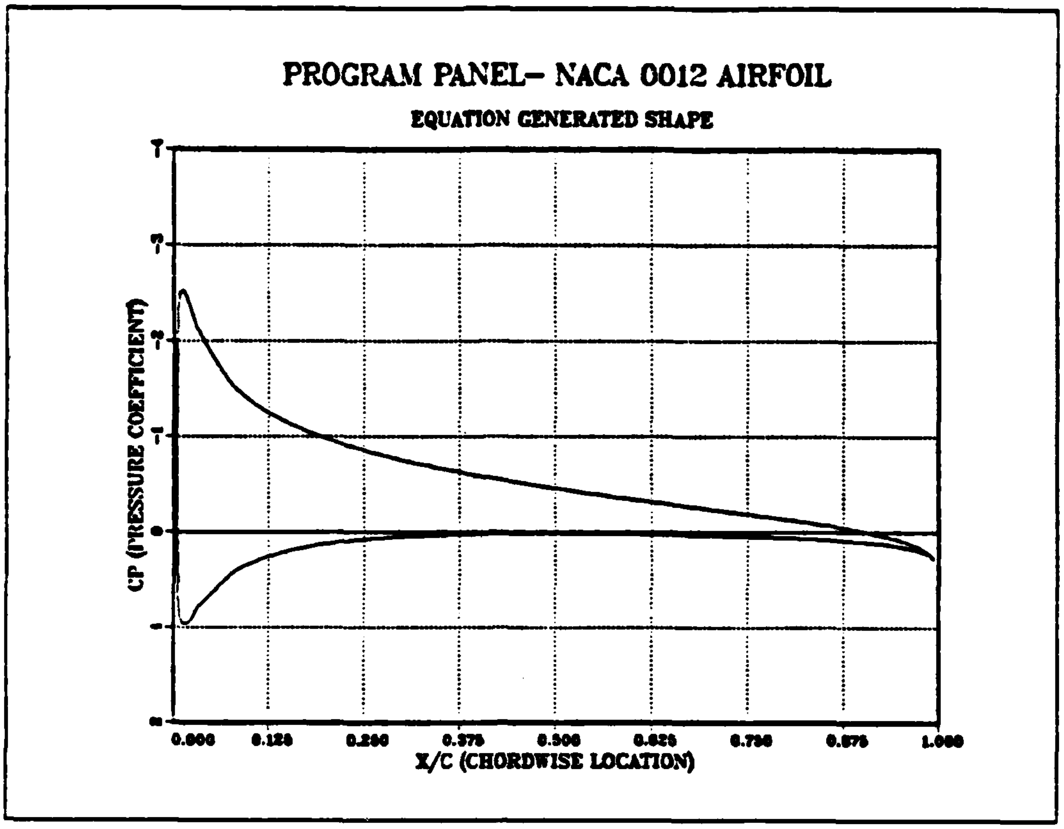


**Figure 31. Program PANEL- Surface Pressure Distribution for NACA 0012 Generated by the Internal Equation**

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The pressure distribution is for the NACA 0012 airfoil defined by the internal equation using 28 surface points, at an angle of attack of six degrees. The results of the program run are repeated below. A slight difference is noted between the plots and the values obtained. This is due largely to the difference in the number of data points used and the spline interpolation used by the plotting routine.

```
DATA FILE:  PPRESS.DAT
ANGLE OF ATTACK IN DEGREES =     6.000
   CD = 0.00721    CL = 0.72235    CM =-0.18377    CMC4 =-0.00398
```

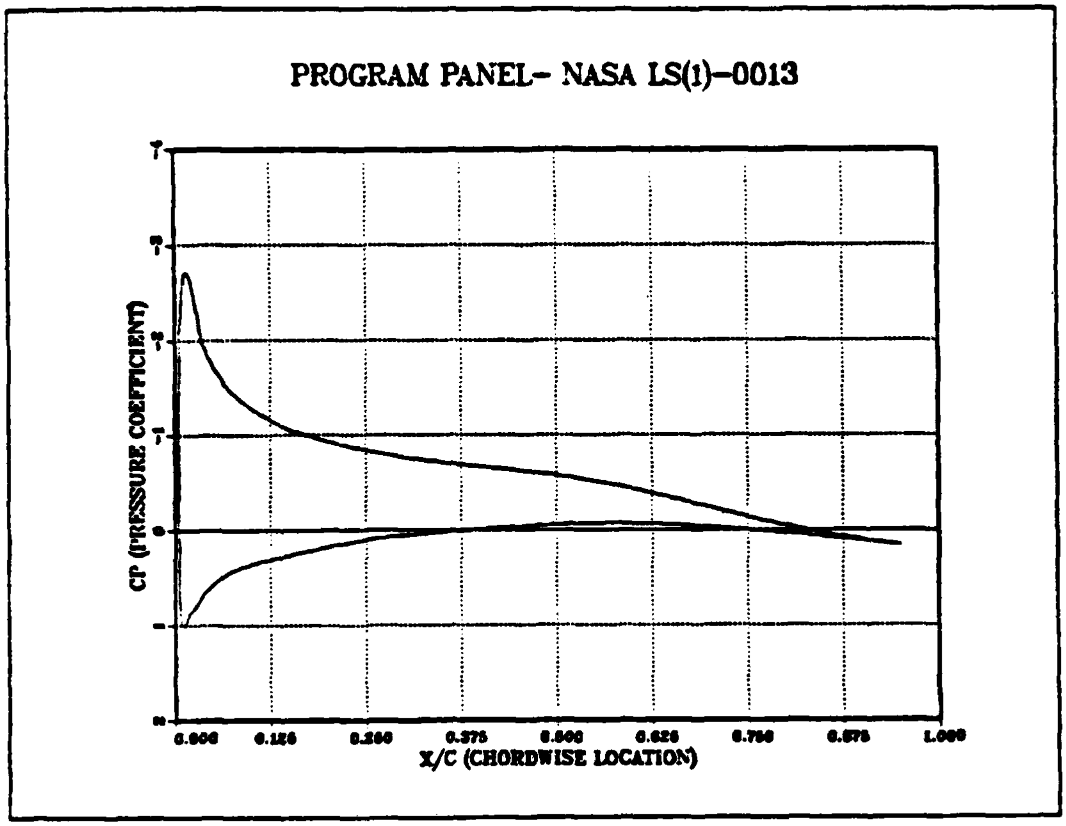


Figure 32. Program PANEL- Surface Pressure Distribution for NASA LS(1)-0013

This figure was generated using the DISSPLA portion of EASYPLOT on the IBM mainframe computer. The pressure distribution is for the NASA LS(1)-0013 airfoil defined by a set of DATA statements containing 28 surface points at an angle of attack of six degrees. The results of the program run are repeated below.

```
DATA FILE:   PPRESS.DAT
ANGLE OF ATTACK IN DEGREES =     6.000

   CD = 0.00324     CL = 0.69366     CM =-0.16505     CMC4 = 0.00750
```

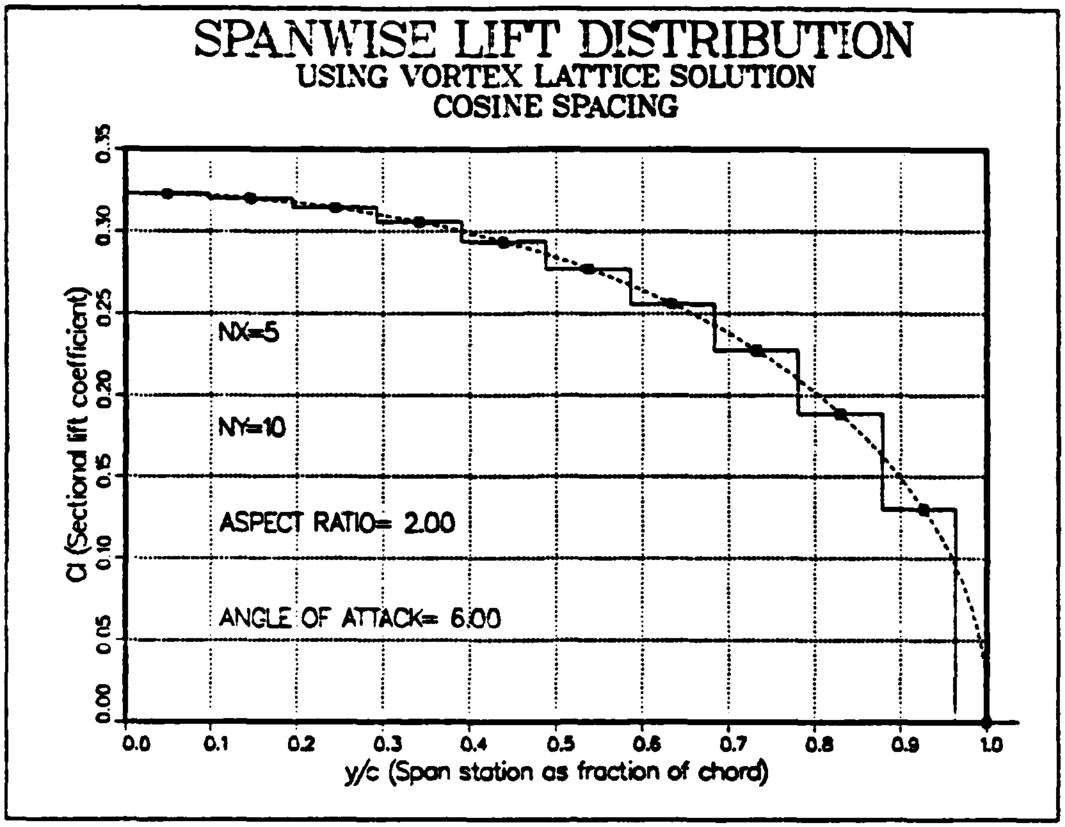


**Figure 33. Program VORLAT- Spanwise Lift Distribution Using Cosine Spacing**

This figure was generated using DISSPLA and running the PLOTSPAN program on the IBM mainframe computer. The spanwise lift distribution is shown for a flat rectangular wing of aspect ratio 2 at an angle of attack of six degrees. The results of the VORLAT program run are repeated below. (The PLOTSPAN program is located on the AERO disk of the IBM mainframe.)

```
** COSINE GRID SPACING **

 NX=  5  NY= 10   ASPECT RATIO =  2.00   ANGLE OF ATTACK =  6.00

CL =       0.25905
CD =       0.0106492
CD/CL2 = 0.1587
CMLE =    -0.055061
XCP =      0.21255
```

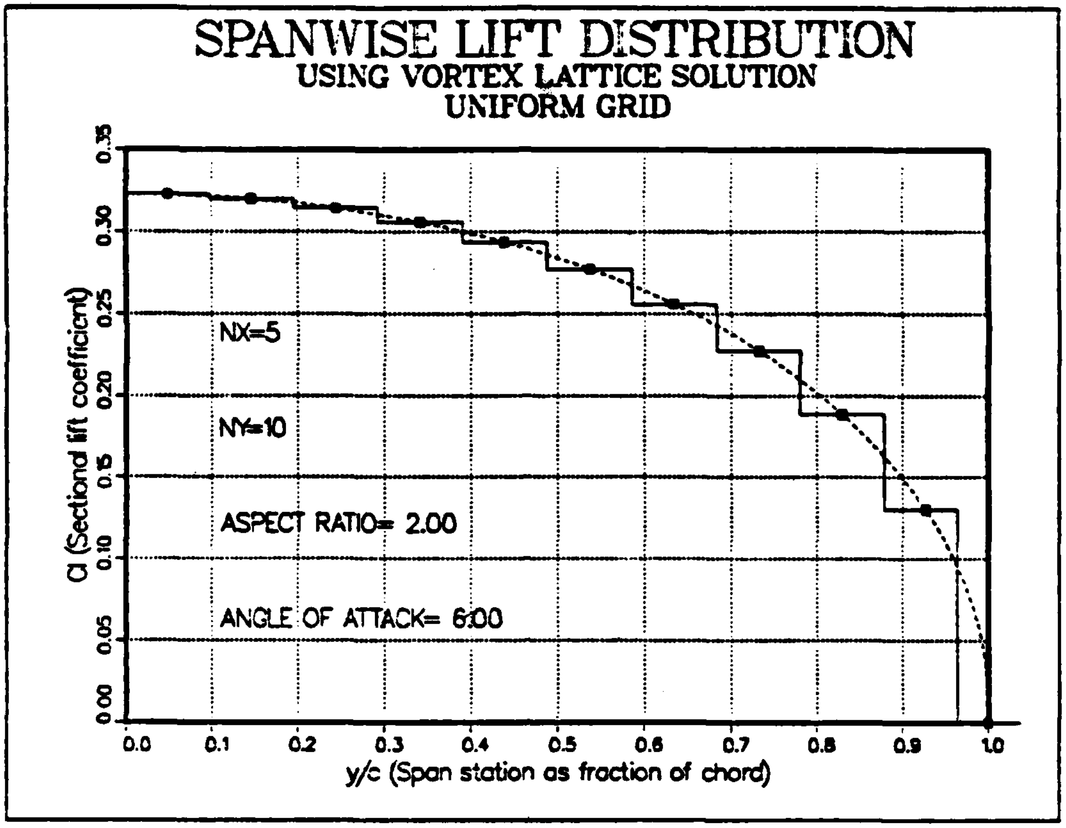


**Figure 34. Program VORLAT- Spanwise Lift Distribution Using Uniform Grid**

This figure was generated using DISSPLA and running the PLOTSPAN program on the IBM mainframe computer. The spanwise lift distribution is shown for a flat rectangular wing of aspect ratio 2 at an angle of attack of six degrees. The results of the VORLAT program run are repeated below.

```
** UNIFORM GRID SPACING **

 NX=  5  NY= 10   ASPECT RATIO =  2.00   ANGLE OF ATTACK =  6.00

CL =       0.25711
CD =       0.0105673
CD/CL2 = 0.1598
CMLE =   -0.054301
XCP =      0.21119
```

## LIST OF REFERENCES


1. Moran, Jack, *An Introduction to Theoretical and Computational Aerodynamics*, pp.32-153, John Wiley & Sons, Inc., 1984.

2. Anderson, John D., *Fundamentals of Aerodynamics*, pp.164-267, McGraw-Hill, Inc., 1984.

3. Zucker, R. D., *Aerodynamic Analysis-A Set of Course Notes on Current Aerodynamic Analysis*, Naval Postgraduate School, Monterey, CA, 1977.

4. Clancy, L. J., *Aerodynamics*, pp.170-199, John Wiley & Sons, Inc., 1975.

5. Maron, M. J., *Numerical Analysis*,, 2d ed., pp.71-74, Macmillian Publishing Company, 1987.

6. Hess, J. L., andSmith, A. M. O., "Calculation of Potential Flow About Arbitrary Bodies," *Prog. Aeronaut. Sci.*, v. 8., pp.1-138, 1966.

7. Lopez, M. L., Shen, C. C., and Wasson, N. F., *A Theoretical Method for Calculating the Aerodynamic Characteristics of Jet-Flapped Wings, Vol.I - The Elementary Vortex Distribution Jet-Wing Lifting-Surface Theory. Vol.II - EVD Jet-Wing Computer Program User's Manual*, Douglas Aircraft Company Report MDC J5519(May 1973), Second Edition 1977.

8. Soderman, A. P., *A Three-Dimensional, Finite Element Lifting-Surface Computer Program, Its Utilization and Underlying Theory*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1976.

9. Digital Equipment Corporation, *VAX EDT Reference Manual*, AA-Z300A-TE, September 1984.

10. Harvey, Neil E., *User's Guide to MVS at NPS* , Technical Note MVS-01, Naval Postgraduate School W. R. Church Computer Center, Monterey, CA, May 1988.

11. International Business Machines Corporation, *MVS/370 JCL User's Guide*, GC28-1349-3, 4th ed., August 1987.

12. Mar, Dennis R., *Magnetic Tape Usage Under MVS at NPS*, Technical Note MVS-07, Naval Postgraduate School W. R. Church Computer Center, Monterey, CA, October 1987.

13. White, S. M., *Class Project AE-3501*,, Naval Postgraduate School, Monterey, CA, 1985.

14. Kretzmann, Jane, *User's Guide to VM/CMS at NPS* , Technical Note VM-01, Naval Postgraduate School W. R. Church Computer Center, Monterey, CA, January 1988.

15. International Business Machines Corporation, *Virtual Machine/System Product CMS Command Reference*, SC19-6209-4, 5th ed., December 1986.

16. Digital Equipment Corporation, *VAX FORTRAN User's Guide*, AA-D035D-TE, September 1984.

17. Digital Equipment Corporation, *Programming in VAX FORTRAN*, AA-D034D-TE, September 1984.

18. NASA Technical Memorandum 4003, *Low-Speed Wind-Tunnel Results for Symmetrical NASA LS(1)-0013 Airfoil*, by J. C. Ferris, R. J. McGhee, and R. W. Barnwell, June 1987.

19. Hough, Gary. R., "Remarks on Vortex-Lattice Methods," *Journal of Aircraft*, v. 10, no. 5, pp. 314-316, May 1973.

# INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22304-6145 | 2 |
| 2. | Library, Code 0142<br>Naval Postgraduate School<br>Monterey, CA 93943-5002 | 2 |
| 3. | E. R. Wood, Chairman, Code 67<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 2 |
| 4. | Professor M. F. Platzer, Code 67Pl<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 2 |
| 5. | Professor R. M. Howard, Code 67Ho<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 2 |
| 6. | Professor J. V. Healey, Code 67He<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 2 |
| 7. | Professor G. H. Lindsey, Code 67Li<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 1 |
| 8. | Professor R. D. Zucker, Code 67Zu<br>Department of Aeronautics and Astronautics<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 1 |
| 9. | Commandant (G-PTE-1)<br>U. S. Coast Guard<br>2100 Second St. S.W.<br>Washington, DC 20593-0001 | 2 |

10. Commandant (G-EAE-4) 2
U. S. Coast Guard
2100 Second St. S.W.
Washington, DC 20593-0001

11. Mr. O. D. Wells, MS 3504 1
C-17 Aerodynamics Group
Douglas Aircraft Company
3855 Lakewood Blvd.
Long Beach, CA 98046-0000

12. LT J. A. Campbell 1
9800 Broadmoor Terrace
Upper Marlboro, MD 20772-0000